

International Micro Air Vehicle Conference and Flight Competition 2017

IMAV2017

Proceedings

J.-M. Moschetta, G. Hattenberger, H. de Plinval

organized by
ISAE-SUPAERO – ENAC – ONERA

September 18–22, 2017, Toulouse, France

Preface

This volume contains the papers presented at IMAV2017: International Micro Air Vehicles Conference and Flight Competition 2017.

The International Micro Air Vehicle Conference and Competition is a yearly event that aims at fostering key technologies for the development of micro-air vehicles. It combines a scientific conference and a flight competition intended to all research groups around the world. After Delft (2014), Aachen (2015) and Beijing (2016), this year edition was held in Toulouse, France, from September 18th to 21nd, 2017.

This edition was, again, an occasion to put forward the methodological and technological advances in this rapidly growing field. The impressive variety of application cases, related needs, and breakthroughs has been represented through the papers gathered in this volume. These papers address many challenges, among which one may mention:

- Conception for new configurations, more efficient, easier to manufacture, bio-inspired,...
- Aerodynamics studies to reduce consumption or to improve drones performances, or to perform finer simulations
- Aeroacoustics studies to reduce the noise produced by drones
- Human machine cooperation, a domain where impressive progress is ongoing, which helps to better understand the human brain, and reduce the operator workload
- Improving the control and guidance designs, in particular to allow multi-vehicles cooperation, better trajectory tracking, and wind gusts rejection, collision avoidance, consumption reduction, or load carrying
- Navigation, the capability to localize the drone in complex environments, with a degraded set of sensors, in particular with the use of few information, for example vision only
- Artificial intelligence, decision making in uncertain contexts, which helps the embedded system taking relevant and safe decision in dynamic environments
- Atmospheric wind measurements

This volume shows great achievements in these various areas. Among those high quality papers, the program committee would like to mention that the paper entitled "Quad-thopter: Tailless Flapping Wing Robot with 4 Pairs of Wings" by Christophe De Wagter, Matej Karasek and Guido de Croon have been nominated for the *Best Paper Award*.

The scientific and organization committees is grateful to the support of our sponsors: Région Occitanie, Toulouse Métropole, Parrot, Mathworks, UIMM, Fondation ISAE-Supaéro and to the contribution of our partners: Toulouse ESOF 2018, Airborne Concept, Mairie de Cugnaux, Préfecture de la Haute-Garonne, DSAC-Sud, Aéroport de Francazal.

September 2017
Toulouse

Jean-Marc Moschetta (ISAE-SUPAERO)
Gautier Hattenberger (ENAC)
Henry de Plinval (ONERA)

Program Committee

Yves Briere	ISAE, France
Pascual Campoy	UPM , Spain
Ben M. Chen	National University of Singapore, Singapore
Christelle Cumer	ONERA, France
Guido De Croon	TU Delft, The Netherlands
Henry de Plinval	ONERA, France
Christophe De Wagter	TU Delft, The Netherlands
Gautier Hattenberger	ENAC, France
Thierry Jardin	ISAE, France
Simon Lacroix	LAAS/CNRS, France
Pascal Morin	ISIR, France
Jean-Marc Moschetta	ISAE, France
Sebastien Prothin	ISAE, France
Mark Reeder	Air Force Institute of Technology, USA
Bart Remes	TU Delft, The Netherlands
Sergey Serokhvostov	MIPT DAFE, Russia
Sergey Shkarayev	University of Arizona, USA
Yoko Watanabe	ONERA, France
Simon Watkins	RMIT, Australia

List of Papers

Session SA1: Aerodynamics and flow control

Qualitative Investigation of the Dynamics of a Leading Edge Control Surfaces for Micro Air Vehicle Applications 1
Ashim Panta, Phred Petersen, Simon Watkins, Matthew Marino, Abdulghani Mohamed and A Fisher

Aerodynamic design of a Martian micro air vehicle 9
Thibault Désert, Jean-Marc Moschetta and Hervé Bezard

Study of ducted fans interference for copter type multirotor UAV/RPAS 17
Kirill Stremousov and Maxim Arkhipov

Session SB1: Control designs and analysis for MAVs

A numerical approach for attitude control of a quadrotor 23
Huu-Phuc Nguyen, Jérôme De Miras, Ali Charara and Stéphane Bonnet

Application of a switching control strategy to extract energy from turbulence by a UAV 29
Yves Briere, Federico Pasquali and Nikola Gavriovic

Prioritized Control Allocation for Quadrotors Subject to Saturation 37
Ewoud Smeur, Daan Höppener and Christophe de Wagter

Session SA2: Aero-acoustics investigations

Aeroacoustics investigation on nano coaxial rotor in hover 44
Zhen Liu, Chen Bu, Xiangxu Kong and Dong Yang

Reducing the noise of Micro-Air Vehicles in hover 51
Ronan Serre, Vincent Chapin, Jean-Marc Moschetta and Hugo Fournier

Application of Lattice Boltzmann Method to some challenges related to Micro Air Vehicles 60
Nicolas Gourdain, Thierry Jardin, Ronan Serre, Sébastien Prothin and Jean-Marc Moschetta

Session TA1: Control designs and analysis

Controller Tuning Strategy for Quadrotor MAV Carrying a Cable-suspended Load 89
Nestor Alonso Santos Ortiz, Sylvain Durand, Renaud Kiefer and Edouard Laroche

Landing and Take-off on/from Sloped and Non-planar Surfaces with more than 50 Degrees of Inclination 97
Marco Tognon and Antonio Franchi

Flight Simulation of a MAKO UAV for Use in Data-Driven Fault Diagnosis 103
Elgiz Baskaya, Murat Bronz and Daniel Delahaye

Incremental Nonlinear Dynamic Inversion and Multihole Pressure Probes for Disturbance Rejection Control of Fixed-wing Micro Air Vehicles 111
Elisabeth van der Sman, Ewoud Smeur, Bart Remes, Christophe De Wagter and Qiping Chu

Session TB1: Navigation strategies and the use of vision

Human-Robot Cooperation in Surface Inspection Aerial Missions 121
Martin Molina, Pedro Frau, Dario Maravall, Jose Luis Sanchez-Lopez, Hriday Bavle and Pascual Campoy

An Intelligent Unmanned Aircraft System for Wilderness Search and Rescue 129
Huai Yu, Shijie Lin, Jinwang Wang, Kaimin Fu and Wen Yang

A honeybee's navigational toolkit on Board a Bio-inspired Micro Flying Robot 136
Erik Vanhoutte, Franck Ruffier and Julien Serres

Loosely Coupled Stereo Inertial Odometry on Low-cost System 143
Haochih Lin and François Defaÿ

Session SA3: Novel designs for MAVs

Investigation on Natural Frequency and Fuselage Effect for Small UAVs Lateral Motion 149
Mostafa El-Salamony and Sergey Serokhvostov

Team MAVion entry in the IMAV'17 outdoor challenge – A tail-sitting trajectory-tracking uUAV 156
Leandro R. Lustosa, Jacson Miguel Olszanecki Barth, Jean-Philippe Condomines, Francois Defay and Jean-Marc Moschetta

Simulation and Control of a Tandem TiltWing RPAS Without Experimental Data 162
Yannic Beyer, Thomas Krüger, Arne Krüger, Meiko Steen and Peter Hecker

Session SB3: Wind measurements or rejection

Using MAVs for Atmospheric Wind Measurements: Opportunities and Challenges 170
Simon Watkins, Mohamed Abdulghani, Sam Prudden, Matthew Marino, Reece Clothier, Alex Fisher and Ashim Panta

Bio-inspired Wind Field Estimation-Part 1: AoA Measurements Through Surface Pressure Distribution . 175
Nikola Gavrilovic, Murat Bronz, Jean-Marc Moschetta, Emmanuel Benard and Philippe Pastor

Developing a Stable Small UAS for Operation in Turbulent Urban Environments 184
Rohan Gigacz, Abdulghani Mohamed, Pakorn Poksawat, Simon Watkins and Ashim Panta

Session TA2: Multiple vehicles cooperation

Collision Avoidance of multiple MAVs using a multiple Outputs to Input Saturation Technique 190
Corentin Chauffaut, Laurent Burlion, François Defaÿ and Henry de Plinval

A Combined Approach for 3D Formation Control in a Multi-UAV System using ROS 196
Rafael Braga, Alexandre Carlos Brandao Ramos, Roberto Claudino Da Silva and Félix Mora-Camino

Formation flight of fixed-wing aircraft by employing guidance vector fields 203
Hector Garcia De Marina and Gautier Hattenberger

EDURA: an Evolvable Demonstrator for Upset Recovery Approaches with a 3D-printed Launcher 209
Torbjørn Cunis and Murat Bronz

Session TB2: Novel design methodologies for MAVs

Optimization of Energy Consumption for Quadrotor UAV 215
Fouad Yacef, Nassim Rizoug, Omar Bouhali and Mustapha Hamerlain

Development and Design Methodology of an Anti-Vibration System on Micro-UAVs 223
Zhenming Li, Mingjie Lao, Swee King Phang, Mohamed Redhwan Abdul Hamid, Kok Zuea Tang and Feng Lin

Quick aerodynamic design of micro air vehicles 229
Atem Kislovskiy and Viktor Vyshinsky

Copter size minimization for the IMAV-2017 competition in Record breaking session 235
Boris Makaev and Sergey Serokhvostov

Session SA4: Specific MAVs designs: flapping and folding wings, and bio-inspired designs

Bond Graph based design tool for a passive rotation flapping wing 242
Le Anh Doan, Sebastien Grondel, Eric Cattan and Christophe Delebarre

Quad-thopter: Tailless Flapping Wing Robot with 4 Pairs of Wings. 249
Christophe De Wagter, Matej Karasek and Guido de Croon

Analysis of Folding Wing Rolling Moment 257
N Krajangsawasdi, T Pantuphag, S Catteeyothai and Chinnapat Thipyopas

Session SB4: Image processing developments

Development of Vision Based Navigation for Micro Aerial Vehicles in Harsh Environment 263
Hailong Qin, Yingcai Bi, Feng Lin and Ben M Chen

Efficient Global Indoor Localization for Micro Aerial Vehicles 270
Volker Strobel, Roland Meertens and Guido de Croon

3D Reconstruction of Complex Structures with Online Profiling and Adaptive Viewpoint Sampling 278
Abdullah Abduldayem, Dongming Gan, Lakmal Seneviratne and Tarek Taha

An Automated Rapid Mapping Solution Based on ORBSLAM and Agisoft Photoscan API 286
Markus Bobbe, Alexander Kern, Yogesh Khedar, Simon Batzdorfer and Ulf Bestmann

Poster session

Investigation on Boundary layer Ingestion Propulsion for UAVs 293
Leonid Teperin, Mostafa El-Salamony, Ahmed Moharam and Moamen Shehata

Multidisciplinary optimization of a MAV propeller for noise reduction 301
Franck Boyer, Arnaud Drapier, Yann Merillac, Cyril Nana and Ronan Serré

Propeller Performance Calculation for Multirotor Aircraft at Forward Flight Conditions and Validation with Wind Tunnel Measurements 307
Christian Molter and Po Wen Cheng

Author Index

Abdul Hamid, Mohamed Redhwan 223
Abduldayem, Abdullah 278
Abdulghani, Mohamed 170
Arkhipov, Maxim 17

Baskaya, Elgiz 103
Batzdorfer, Simon 286
Bavle, Hriday 121
Benard, Emmanuel 175
Bestmann, Ulf 286
Beyer, Yannic 162
Bezard, Hervé 9
Bi, Yingcai 263
Bobbe, Markus 286
Bonnet, Stéphane 23
Bouhali, Omar 215
Boyer, Franck 301
Braga, Rafael 196
Briere, Yves 29
Bronz, Murat 103, 175, 209
Bu, Chen 44
Burlion, Laurent 190

Campoy, Pascual 121
Cattan, Eric 242
Catteeyothai, S 257
Chapin, Vincent 51
Charara, Ali 23
Chauffaut, Corentin 190
Chen, Ben M 263
Cheng, Po Wen 307
Chu, Qiping 111
Claudino Da Silva, Roberto 196
Clothier, Reece 170
Condomines, Jean-Philippe 156
Cunis, Torbjørn 209

De Croon, Guido 270, 249
De Miras, Jérôme 23
de Plinval, Henry 190
De Wagter, Christophe 37, 111, 249
Defay, Francois 156
Defaÿ, François 143, 190
Delahaye, Daniel 103
Delebarre, Christophe 242
Doan, Le Anh 242

Drapier, Arnaud 301
Durand, Sylvain 89
Désert, Thibault 9

El-Salamony, Mostafa 149, 293

Fisher, A 1
Fisher, Alex 170
Fournier, Hugo 51
Franchi, Antonio 97
Frau, Pedro 121
Fu, Kaimin 129

Gan, Dongming 278
Garcia De Marina, Hector 203
Gavrilovic, Nikola 175
Gavriovic, Nikola 29
Gigacz, Rohan 184
Gourdain, Nicolas 60
Grondel, Sebastien 242

Hamerlain, Mustapha 215
Hattenberger, Gautier 203
Hecker, Peter 162
Höppener, Daan 37

Jardin, Thierry 60

Karasek, Matej 249
Kern, Alexander 286
Khedar, Yogesh 286
Kiefer, Renaud 89
Kislovskiy, Atem 229
Kong, Xiangxu 44
Krajangsawasdi, N 257
Krüger, Arne 162
Krüger, Thomas 162

Lao, Mingjie 223
Laroche, Edouard 89
Li, Zhenming 223
Lin, Feng 223, 263
Lin, Haochih 143
Lin, Shijie 129
Liu, Zhen 44
Lustosa, Leandro R. 156

Makaev, Boris 235
Maravall, Dario 121
Marino, Matthew 1, 170
Meertens, Roland 270
Merillac, Yann 301
Mohamed, Abdulghani 1, 184
Moharam, Ahmed 293
Molina, Martin 121
Molter, Christian 307
Mora-Camino, Félix 196
Moschetta, Jean-Marc 9, 51, 60, 156, 175

Nana, Cyril 301
Nguyen, Huu-Phuc 23

Olszanecki Barth, Jacson Miguel 156

Panta, Ashim 1, 170, 184
Pantuphag, T 257
Pasquali, Federico 29
Pastor, Philippe 175
Petersen, Phred 1
Phang, Swee King 223
Poksawat, Pakorn 184
Prothin, Sébastien 60
Prudden, Sam 170

Qin, Hailong 263

Ramos, Alexandre Carlos Brandao 196
Remes, Bart 111
Rizoug, Nassim 215
Ruffier, Franck 136

Sanchez-Lopez, Jose Luis 121
Santos Ortiz, Nestor Alonso 89
Seneviratne, Lakmal 278
Serokhvostov, Sergey 149, 235
Serre, Ronan 51, 60
Serres, Julien 136
Serré, Ronan 301
Shehata, Moamen 293
Smeur, Ewoud 37, 111
Steen, Meiko 162
Stremousov, Kirill 17
Strobel, Volker 270

Taha, Tarek 278
Tang, Kok Zuea 223
Teperin, Leonid 293
Thipyopas, Chinnapat 257
Tognon, Marco 97

van der Sman, Elisabeth 111
Vanhoutte, Erik 136
Vyshinsky, Viktor 229

Wang, Jinwang 129
Watkins, Simon 1, 170, 184

Yacef, Fouad 215
Yang, Dong 44
Yang, Wen 129
Yu, Huai 129

Qualitative Investigation of the Dynamics of a Leading Edge Control Surfaces for MAV Applications

A Panta*, Petersen P, Marino M, Watkins S, Fisher A and Mohamed A
RMIT University, Melbourne Australia

ABSTRACT

Conventional control surfaces mounted on wing trailing edges actuated with commercially available servos have not been able to achieve sufficient control authority and rapidity to keep small MAVs flying straight and level in turbulent flow. Non-conventional leading edge control surfaces are investigated as an alternative actuation solution with the potential to enhance control authority and rapidity. In this study, flow visualization of leading edge control surface revealed that higher deflection rates delayed flow separation and this is expected to enhance control forces. Higher actuation rates produced dominant leading edge vortices and hence a transient lift enhancement over the airfoil. Lift spikes from high rate actuations could be exploited to compensate for the high frequency perturbations from gusts.

1 INTRODUCTION

Small Micro Air Vehicles (MAVs) are generating high level of interests in the unmanned sector of aviation because of the diverse range of reconnaissance, surveillance and package delivery missions these lightweight systems can fulfill. However their miniature size and flight environment introduces a range of flight challenges primarily due to the relatively high levels of turbulence present at low altitudes where MAVs operate [1–6]. High frequency energy content in turbulence has the ability to rapidly accelerate and rotate these lightweight systems [7, 8]. When attempting to attenuate these deleterious effects of turbulence the actuation rate of conventional fixed wing control surfaces has been found insufficient to adequately compensate for the disturbance inputs. This is due to the relatively higher aircraft frequency response required by MAVs, coupled with the limited control authority and actuation power. Existing control surface placement and turbulence response systems do not have sufficient power and rapidity to overcome perturbations in turbulence to a level where MAVs can fly steady in urban environment [9] .

A range of passive and active methods have been explored to address this issue of poor attitude control in MAVs. Passive methods involve the aircraft's natural ability to produce the aerodynamic forces to achieve stability, through design features of the aircraft (eg., wing sweep, dihedral etc). Existing literature show that these techniques can only attenuate low frequencies of perturbations while limiting maneuverability and agility [10]. Active methods in contrast refers to the use of a control system, that goes through a Sense (detect turbulence ahead of the aircraft), Plan (consider desired control surface deflection ahead of time) and Act (aerodynamic actuation) cycle [11], see Figure 1. It is near impossible to manually fly these aircraft in turbulence [12]. Many MAVs require control input rates higher than 25 Hz [13], which is beyond the bandwidth of human operators [14]. Employment of an active attitude control system is therefore vital as a microcontroller can provide higher input control rates than human pilots.

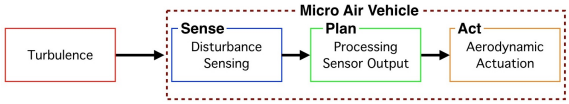


Figure 1: Control system's process for controlling MAV in turbulence, adopted from [15]

Despite active turbulence mitigation techniques such as Phased Advanced Sensing [9] and Real Time Pressure Sensing [16] showing promising results in the Sense and Plan component of the SPA cycle, it was found that conventional designs and control surfaces (i.e. mounted on the trailing edges of flying surfaces) could not achieve sufficient control authority and response to keep small unmanned craft flying straight and level in turbulence [9]. This is primarily due to the lower than required speed of mechanical actuator and small control surface of MAVs.

Potential solutions for increasing the control authority lay in the use of control surfaces that are hinged on the leading edges of wings. These provide an unstable hinge moment where the fluid is driving the actuator (forcing a passive moment), rather than resisting against it, contrary to the situation with conventional control surfaces. Concerns with the unstable nature of such Leading Edge Control Surfaces (LECS) mean that they have not been used on larger manned aircraft. However the relatively low loads on fixed wing MAV, coupled with the requirement to operate controls at much higher frequencies than manned aircraft, make them potentially

*Email address(es): ashim.panta@rmit.edu.au

useful for rapid maneuverability and turbulence rejection. As a result it might be possible for leading edge devices to be used more freely for MAV applications and could provide an insight and solutions to the controllability issues of MAVs.

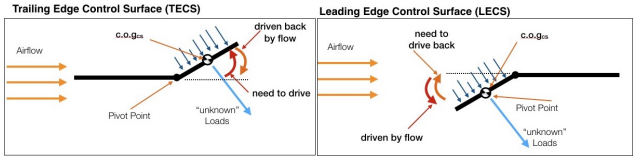


Figure 2: Free body diagram of trailing edge (left) and leading edge (right) control surfaces

Trailing Edge Control Surfaces (TECS) need power from an actuator to rotate a control surface from a neutral position to a deflected position in order to generate an aerodynamic control, see Figure 2. At this deflected position there is a restoring moment from the flow, which could be used to drive the surface back to its neutral or un-deflected position, thus potentially no actuator power is required for this part of the motion. LECS can use the flow to drive the control surface from its un-deflected position to a predetermined deflected position (i.e. there is no need for servo actuation for this part of the motion) but an actuator is needed to bring the control surface back to a neutral position and to hold it in position. So there are potential disadvantages present with the proposed solution. The complex dynamics of a LECS for low Reynolds Number (Re) ($\leq$150,000) flight is not well understood in the current body of literature. Thus the feasibility of using such system for MAV controls (to improve control authority and response) cannot be dictated without further investigation.

1.1 Low Reynolds Number Leading Edge Aerodynamics

1.1.1 Static Effects

Though the use of leading edge control surfaces for primarily control is not common, the use of leading edge devices to enhance performance is a well explored concept. Airfoil performance may be enhanced by management of boundary layer using leading surface roughness, suction or blowing. Adverse effects of separation bubbles and bubble burst (which can dominate the flows at low Re) can be overcome by various means. Transition control can be done by advancing the transition or maintaining the laminar flow both of which can suppress flow separation bubble effects [17]. At low Reynolds Numbers, flow separation occurs near the leading edge, thus it is common to employ a leading edge flow control mechanisms in order to main attached flow at high angles of attack. A thorough review of flow control mechanisms is presented by [18] to control the leading edge vortices on delta wing aircraft. Along with blowing, suction and unsteady excitation techniques the author suggested the use of leading edge control surfaces.

A study by [17] found statically deflected leading edge flaps (much like Kruger flap) enhanced aerofoil performance by augmenting lift and limiting drag at certain angle of attack. These flaps acted like a transition device preventing the formation of separation bubbles. A range of accepted flow control methods for low Re was reviewed and the most promising method was to passively design such that the (leading edge curvature, camber and thickness) severity of adverse pressure gradient forces transition to the desired location. This study was inspired by the findings of leading edge control surfaces (flaps) in natural flyers much like the study by [19]. It was found that at Re of 40000 - 120000 the addition of a leading edge flaps showed distinct performance enhancement at angle of attack ($\geq 20^0$). The leading edge flaps were found to increase the baseline airfoil's lift by up to $C_l = 0.52$. Further use of leading edge flaps as flow control mechanism can be found in [17–25].

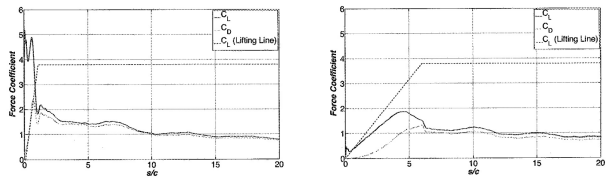
1.1.2 Dynamic Effects

MAVs must be capable of executing agile and aggressive maneuvers at highly unsteady dynamic conditions. In unsteady flow, lift generated is contributed from circulatory and non circulatory components. Circulatory components include Leading Edge Vortices (LEV) and bound circulation while non-circulatory forces are brought about the wing's acceleration and added mass. Leading edge devices are good to generate vortical structures that have significant contributions to the overall force production on an airfoil. Examples of these are observed in natural flapping flyers such as birds and insects [26]. Thus recent research in the unsteady aerodynamics associated with MAV flights has been increasingly focusing on LEVs. LEV is a physical flow phenomenon that generally occurs during a dynamic stall; i.e., when an airfoil, rapidly pitched up beyond its static stall angle generates a dynamic stall vortex, causing the lift coefficients to increase beyond its maximum value for the un-stalled case. LEVs are well studied in various aerodynamic contexts such as retreating helicopter blades or super maneuverable aircraft [27]. In the unsteady low Re regime the LEV is believed to contribute most to the lift generation [28]. As MAV flight requires rapid controls to maneuver in and around obstacles and to overcome turbulent disturbances, MAV flow physics is highly unsteady. Thus the formation of LEV have a significant influence in providing high lift coefficient for MAV applications [29]. Since rapid pitch or flapping motions occur on a smaller time scale than the development of full stall, this enhancement is exploited by small airborne creatures and is of interest for designing agile MAVs [27, 30].

A LEV generates lift increment through the low pressure region induced from the vortex core on the upper surface of the wing and provides short term enhancement of lift whilst remaining attached to the wing. LEVs are created when the adverse pressure gradient and viscous shear stresses create flow separation which causes a vortex to break away from the leading edge of the airfoil. LEVs can follow along the chord (desired) or break away from the airfoil (detrimental) [31]. At high angles of attack LEVs make significant contribution to the total lift of the wing [32]. Studies by [32] suggested the possibility of generating appreciable pitching and rolling moments for flight control. LEVs were found to shed at increasing rates for increasing Re and angle of attack in low Re [33].

1.1.3 Effects of Actuation Rates

Actuation rates also have a significant influence in unsteady force production over an airfoil. A study on a rapidly pitching flat plate wing found that the starting LEV is more pronounced at higher reduced pitch rates [34]. The lift peak was found to correlate with the maximum size of the LEV before its shedding and downstream convection [35]. A study by [36] also found that fast pitch rate enhanced the forces during the rapid pitch motions. It was found that the development and convection of LEV is linear until eruption. Attached flow from fast pitch correlates to higher force coefficients than slow pitch, where flow is quasi steady. Figure 3 from [36] shows the lift characteristics as a function of pitch rate. Furthermore investigation of a NACA 0015 airfoil pitching constantly at the mid chord by [37] measured time varying pressure drag and moment coefficients as a function of angle of attack. This study also found that higher pitch rates had dramatic positive effects on both the delay of stall and the magnitude of maximum lift coefficients, Figure 4.



(a) Forces Pitching over 1c. (b) Forces Pitching over 6c.

Figure 3: Force coefficients for fast and slow pitch rates [36]

1.2 Objectives

The little work surrounding the low Re aerodynamics of leading edge flaps suggests that these devices improve the airfoil performance. However a comprehensive understanding of the fluid dynamics of a leading edge flap (as opposed to rapid motion of an entire airfoil) and the effects of flap deflection angles and rates on the overall force production in

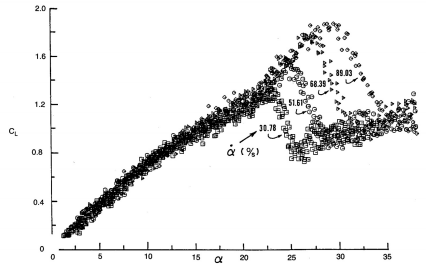


Figure 4: Coefficients of lift at various pitch rates [37]

unsteady cases were not well established. Thus the objective of this paper is to establish qualitative perspective on the flow structures formed behind a LECS. To get an insight into the fluid mechanics associated, flow visualizations for various control surface deflection angles and airfoil angle of attack were investigated for static and dynamic deflections.

2 Experimental Apparatus

2.1 Wind Tunnel Setup

An insert box was manufactured to give nominally 2D flow across the span of a flat plate airfoil in the RMIT Aerospace Wind Tunnel. The tunnel is closed-return and has a hexagonal test section of 2.1m, 1.3m and 1.1m [LxWxH]. The insert box was fully transparent for a controlled 2D airfoil experiment while allowing video data acquisition. The insert was a Plexiglass box of 1m x 0.5m x 0.5m. The flat-plate airfoil used was 1% thick and featured a rounded leading edge and a blunt trailing edge as documented by [38, 39]. Trailing edge geometry is known not to have a strong effect on the lift and drag at low Re. The chord was 135 mm; a size typical of a fixed wing MAV. The flat-plate leading edge control surface was 30% of the chord with a similar thickness. The wing was mounted horizontally across the insert box, slotting into rotating dowels on either ends, see Figure 5. The wing was statically fixed but the leading edge control surface was free to rotate about its hinge point (at 30% of chord from the leading edge). The control surfaces were actuated by RJX 1001 servos (mounted on the wing) using an externally positioned Arduino board. Position feedback of the servo was attained from tapping the servos potentiometer.

2.1.1 Pitching Kinematics

Three different cases of control surface deflections were analyzed; a static deflections and fast and slow deflections based on reduced frequency k, [31]

$$k = \frac{\omega c}{U} \tag{1}$$

Unsteady effects increases with increased reduced frequency. ω is circular frequency (rad/s), U ref is reference velocity (m/s) and c is chord (m). The fast pitch motions were done

at 1200 deg/sec (k=0.14) and slow pitch at (k=0.0017) at 15 deg/sec at Re of 30,000. Typically this involved the wing positioned at a certain angle of attack, and the control surface being accelerated at a constant rate from 0^0 to a maximum of $\pm 30^0$.

2.2 *Flow Visualization*

A 0.15 mm diameter nichrome wire was mounted vertically across the insert box in order to generate multiple smoke filaments. The wire was heated electrically (30V at 0.6 Amps) in order to vaporize a mixture of iron powder and glycerin. A Phantom Miro M310 high-speed camera was mounted from above focusing on the smoke filaments. Recordings were done at 1280 x 720 pixels with sample rate of 1000 fps.

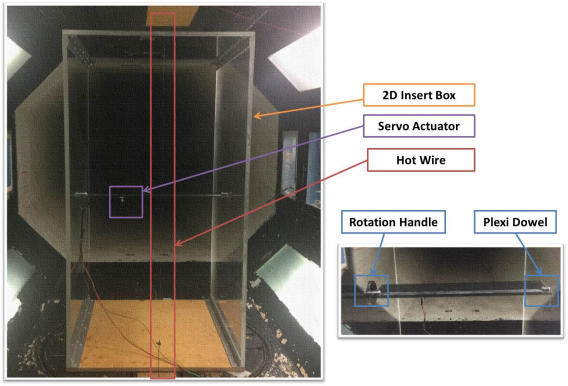


Figure 5: Wind tunnel setup for flow visualization

3 Static Analysis

Figure 6 shows the flow over the airfoil at zero-angle of attack with controls undeflected. Flow features were found to correlate well with existing literature on flat plate airfoils at low Re, including the von Karman vortex sheet seen in Figure 6, due to the blunt trailing edge shape [38, 39].

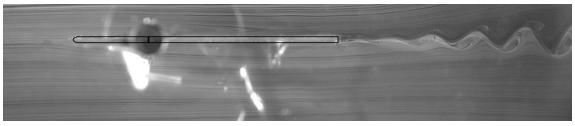


Figure 6: Flow over the wing at zero incidence

For all positive deflections of the control surface (δ), the flow remained attached over the entire lower (pressure) surface for all angles of attack (α) tested. As expected from symmetry, all - δ deflections displayed an attached flow over the upper side for α angles up to 20 degrees. During δ deflections, the flow remained attached on the entire lower surface, up to 5^0 α. At higher incidences thin airfoils are subject to leading edge laminar separation due to the pronounced

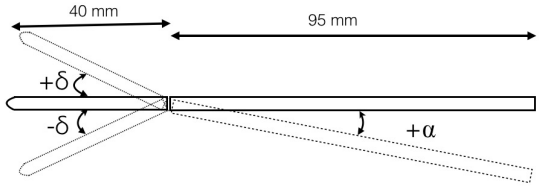


Figure 7: Sign convention, δ = flap deflection angle, α= angle of attack

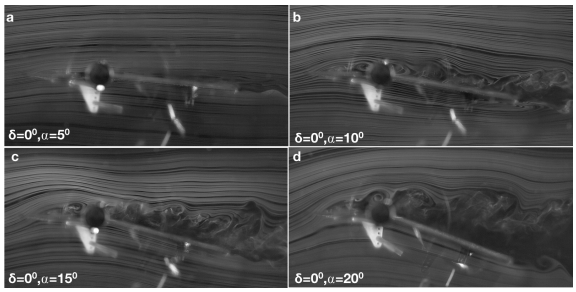


Figure 8: Flow over the wing at various incidence angles

suction peak shortly downstream of the leading edge. This was clearly seen in Figures 8 b-d in which a Laminar Separation Bubble (LSB) formation is visible. The change in pressure distribution due to LSB formations generally lead to decreased lift while increasing drag [40]. LSBs seen in Figures 8 b-d are relatively small in dimension and are considered short" bubbles which do not significantly alter the pressure distribution around the airfoil [41, 42].

4 Dynamic Analysis

Dynamic actuation of the LECS is analyzed in this section and flow patterns are compared with the static cases. A wide range of α's were tested and only the most significant variation in between static and dynamic deflections are presented. Images from the static case in Figure 8 were captured at an arbitrary time when the flow was fully developed over the airfoil. Whereas the images from the dynamic cases presented in Figures 9-12 were captured as the LECS reached the desired deflection angle.

Figure 9 shows the comparison of flow over LECS and wing for the three cases; static, slow actuation and fast actuation. For the cases of α =0^0 rapid actuation significantly changed the location of the stagnation line on the upper part of the control surface, promoted flow attachment on the lower surface and completely removed the flow separation on the lower surface of the airfoil. For the cases of $\alpha = 10^0$ rapid actuation promoted separation on the upper part of the control surface with a reduced region of separated flow over the upper surface of the airfoil. The flow over the upper surface was moderately attached during low δ deflections for $\alpha > 10^0$.

Beyond that, formations of dominant Leading Edge Vortices (LEV) were visible at the fastest deflection rate (Figure 9). The flow on the upper surface was found to be of complex nature and thus requires quantitative investigation of the pressure/forces on the airfoil. Thus the actuation rate of the LECS significantly influenced how the flow behaves and is expected to significantly alter the dynamic control forces on both control surface and airfoil.

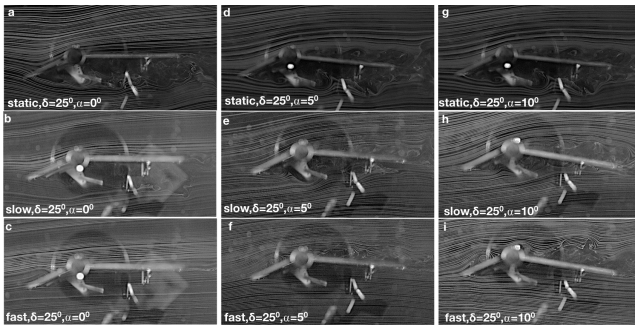


Figure 9: Comparison of flow during a static case and actuation rates of k=0.0017 (slow) and k=0.14 (fast) during negative flap deflections; a-c: increasingly actuation rate for 0^0 angle of attack (AoA) , d-f:increasingly actuation rate for 5^0 AoA, g-i:increasingly actuation rate for 10^0 AoA

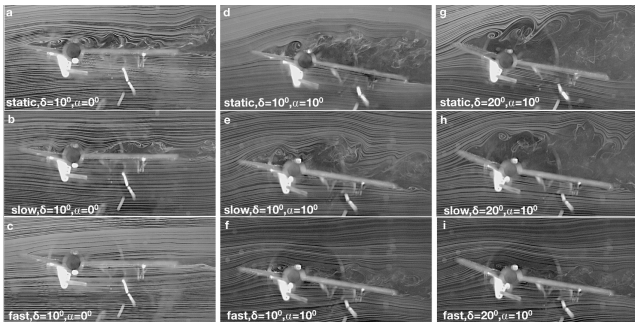


Figure 10: Comparison of flow during a static case and actuation rates of k=0.0017 (slow) and k=0.14 (fast) during positive flap deflections; a-c: increasingly actuation rate for 0^0 AoA, d-f:increasingly actuation rate for 5^0 AoA, g-i:increasingly actuation rate for 10^0 AoA

Observing both Figure 9 and 10, it is clear that both slow and fast actuation cases reduced the amount of flow separation on both the pressure and suction surfaces. The reduced flow separations on both surfaces of the airfoil were more noticeable during the fast actuation cases. An investigation of the effects of rapid actuation rates on a conventional TECS configuration by [43] also found that flow separation was delayed with faster deflection rates. Direct force measurements in a water tunnel demonstrated that the total lift coefficient responded immediately upon initiating the deflection of the control surface [43]. This implies that the attached flow found in this qualitative investigation suggests more lift production. However, these benefits reduce once the flow develops back to its common static state.

Whilst fast pitch was found to produce more dominating LEV, slow pitch was found to be largely separated and featured small LEV sheds (Figure 9). During a fast linear pitch classical LEV is formed, dominating the flow. It is hypothesized that pitch component of a shear layer aids the LEV formation. At slow pitch rates the flow was generally separated and dominance of LEV was not seen. Development of upper surface flow and LEV was strongly correlated with the kinematics of the leading edge, suggesting that local angle of attack at leading edge is of high significance in unsteady pitching motions. Investigations of the effect of pitch rate on the LEV size were done by [29, 44, 45]. In these investigations it was found that increasing the pitch rate delayed the formation of LEV on the upper surface, and made LEV more compact and stronger. While a fast pitch motion produced classical (dominant) LEV slow pitch motions lead to non dominating LEV structures where flow seemed largely separated with small LEV sheds. These findings correlate well with the qualitative results presented in this paper, Figure 9 and 10.

Results presented in this paper indicate that actuation rate of LECS have a significant influence on the flow structures, especially the LEVs, which can be related to the lift enhancement on the wing. This implies that higher deflection rates produce departures from quasi-steady response due to the lift contributions from the circulatory components, enhancing force production during the deflection phases, while slow deflection rates can be expected to be closely quasi-static. Thus it can be concluded that the flow structures are a strong function of pitch rates and that higher pitch rates means higher the angle of attack before the beginning of flow separation and more energetic suction peak. Thus higher deflection rates have significant effects on both the delay of stall momentarily and the magnitude of maximum lift coefficients.

4.1 Flow Characteristics over time

Figure 11 displays a time history of the control surface when exposed to smoke flow stream lines. Three separate angle of attack angles are shown. The time sequence displayed is post full deflection to investigate the flow mechanics directly after the leading edge deflection. The formation of LEV initiates as the control surface motion completes and remains in a deflected state. The maximum defection angle of the control surface was 30^0. In all cases the leading edge deflection activates the formation of a LEV. LEVs are seen to grow in size as they convect downstream with the flow. As the angle of attack is increased, the convection rate of the

LEV is more aggressive on the suction surface of the airfoil. The LEVs are seen to grow however this growth is disrupted when the vortex enters into the favourable pressure gradient. This occurrence is more pronounced in the higher angles of attack with the vortex almost non-existent in the final frames. LEV on the upper surface were found to traverse faster with increasing angle of attack, see Figure 11.

Experiments where performed in the same manner with actuating opposite deflection increasing the angle of attack of the airfoil (see Figure 12). In this case a LEV was only produced on the upper surface where suction pressure exists. The bottom surface did not show visible signs of LEVs due to the effect of the favorable pressure gradient. The generation of the vortex was more pronounced relative to the downward deflection presented previously. This suggests significant increases in lift generation. This increase is short lived as the vortex and surrounding flow return back to a steady flow scenario. Short lived LEVs shown here may provide a means to increase the amount of lift which exceeds what could be achieved with conventional trailing edge control surfaces. The fact that the flow is unsteady and returns to a steady state case means that this actuation may only be viable if done at high frequency and allowed to return to the non-deflected state. Further experiments are needed to evaluate this hypothesis however the strong creation of the LEV suggest that the lift gains may be significant enough for further experiments to quantify the time varying changes in lift.

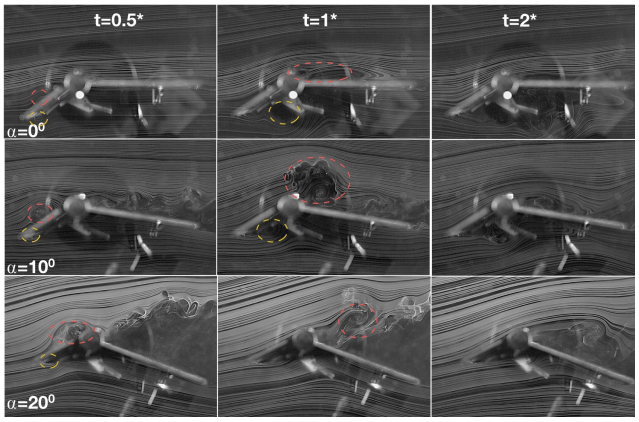


Figure 11: Flow over the wing against characteristic time during negative LECS deflection

5 Conclusion

Through flow visualization experiments it was found that increasing actuation rates on leading edge hinged control surfaces promoted flow attachment on the airfoil, thus could be a potential solution towards achieving high responsiveness and authority required for steady MAV flight in turbulence. The effect of leading edge flap deflection rates at varying

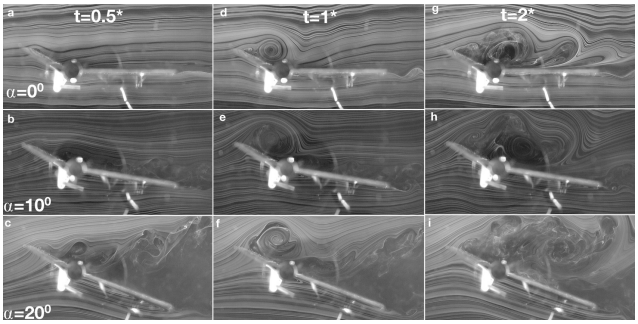


Figure 12: Flow over the wing against characteristic time during positive LECS deflection

wing angles of attack were studied. It was found that higher deflection rates produced a more dominant LEV, which grows significantly as it traverses across the chord of the airfoil. This is expected to provide a significant transient lift increment due to the presence of the low pressure vortex core on the upper surface of the airfoil. The size and development of the LEV at different deflections rates suggest correlation between quantifiable increases of incremental lift and LECS deflection rates.

Furthermore the study suggests that faster LECS actuation leads to greater transient lift production. The return of the LECS to its nominal position will be studied in ongoing experiments. The dynamic influence of a returning LECS (to original position) may uncover other fluid dynamic phenomena which must be accounted for in the overall system. Another quantity which must be accounted for is the effect of control surface mass and how rapid actuation causes secondary forces in line with Newtons third law. The effects of virtual mass are also assumed to have an influence on the force production when LECS are rapidly actuated. Controlling the formation of convection of LEV across the airfoil using a LECS could potentially lead to production of large control forces. It is hypothesized that LECS could hold the key in offering the high-frequency mitigation while conventional control surfaces (hinged at the trailing edge) handle the low frequency disturbances. Further work is needed to understand the generation of these transient pressures and control forces for MAV flight in turbulence. This is part of planned future research (potentially from force and pressure measurements). The potential of this control methodology could serve micro flight applications well as high-frequent response rates are needed to mitigate high frequency perturbations.

6 Acknowledgments

The authors would like to thank the Australian Federal Government and the Defence Science Institute for funding this work, including an Australian Postgraduate Scholarship. The authors gratefully acknowledge Mr Lance Fang for his

assistance during data acquisition and experimental setup.

REFERENCES

[1] Simon Watkins, Juliette Milbank, Benjamin J Loxton, and William H Melbourne. Atmospheric winds and their implications for microair vehicles. *AIAA journal*, 44(11):2591–2600, 2006.

[2] Simon Watkins, Mujahid Abdulrahim, Mark Thompson, Mark Shortis, Reuven Segal, and John Sheridan. An overview of experiments on the dynamic sensitivity of mavs to turbulence. *Aeronautical Journal*, 114(1158):485, 2010.

[3] Darryll J Pines and Felipe Bohorquez. Challenges facing future micro-air-vehicle development. *Journal of aircraft*, 43(2):290–305, 2006.

[4] Abdulghani Mohamed, Kevin Massey, Simon Watkins, and Reece Clothier. The attitude control of fixed-wing mavs in turbulent environments. *Progress in Aerospace Sciences*, 66:37–48, 2014.

[5] GR Spedding and PBS Lissaman. Technical aspects of microscale flight systems. *Journal of Avian Biology*, pages 458–468, 1998.

[6] C Galiński and R Żbikowski. Some problems of micro air vehicles development. *Technical Sciences*, 55(1), 2007.

[7] M Thompson, S Watkins, C White, and J Holmes. Span-wise wind fluctuations in open terrain as applicable to small flying craft. *Aeronautical Journal*, 115(1173):693–701, 2011.

[8] Peter Lissaman. Effects of turbulence on bank upsets of small flight vehicles. In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, page 65, 2009.

[9] A Mohamed, M Abdulrahim, S Watkins, and R Clothier. Development and flight testing of a turbulence mitigation system for micro air vehicles. *Journal of Field Robotics*, 2015.

[10] Mujahid Abdulrahim, Simon Watkins, Reuven Segal, Matthew Marino, and John Sheridan. Dynamic sensitivity to atmospheric turbulence of unmanned air vehicles with varying configuration. *Journal of aircraft*, 47(6):1873–1883, 2010.

[11] Robin Murphy. *Introduction to AI robotics*. MIT press, 2000.

[12] T Chen, R Clothier, A Mohamed, and R Badawy. An experimental study of human performance in controlling micro air vehicles in turbulent environment. In *Fourth Australasian Unmanned Systems Conference*, pages 1–8. Australian Association for Unmanned Systems, 2014.

[13] Song Wang, Yaowei Li, and Wenchang He. Flight attitude estimation for mav based on m-estimation. In *Consumer Electronics, Communications and Networks (CECNet), 2011 International Conference on*, pages 4968–4973. IEEE, 2011.

[14] David A Jenkins, Peter G Ifju, Mujahid Abdulrahim, and Scott Olipra. Assessment of controllability of micro air vehicles. In *Proc. Sixteenth Int. Conf. On Unmanned Air Vehicle Systems*, 2001.

[15] Abdulghani MOhamed. *Phased-Advanced Attitude Sensing and Control for Fixed-Wing Micro Aerial Vehicles in Turbulence*. PhD thesis, RMIT University Melbourne, 2015.

[16] He Shen, Yunjun Xu, and Benjamin T Dickinson. Micro air vehicles attitude control using real-time pressure and shear information. *Journal of Aircraft*, 51(2):661–671, 2014.

[17] NM Bakhtian, Holger Babinsky, ALR Thomas, and GK Taylor. The low reynolds number aerodynamics of leading edge flaps. In *Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit*, pages 8018–8030, 2007.

[18] Ismet Gursul, Zhijin Wang, and Eleni Vardaki. Review of flow control mechanisms of leading-edge vortices. *Progress in Aerospace Sciences*, 43(7):246–270, 2007.

[19] AR Jones, NM Bakhtian, and H Babinsky. Low reynolds number aerodynamics of leading-edge flaps. *Journal of Aircraft*, 45(1):342–345, 2008.

[20] Myong Hwan Sohn, Hyoung Seog Chung, and Jo Won Chang. Control of double-delta-wing vortex by micro leading-edge flap. In *25th AIAA Applied Aerodynamics Conference, Miami*, 2007.

[21] Kosuke Toyoda, Dongyoun Kwak, Masayoshi Noguchi, and Kenichi Rinoie. Aerodynamic interference caused by the inboard leading-edge flap on the outboard area of the cranked arrow wing. *Procedia Engineering*, 99:1642–1646, 2015.

[22] Mohamed Gad-el Hak. Control of low-speed airfoil aerodynamics. *AIAA journal*, 28(9):1537–1552, 1990.

[23] Charles Pitt Ford, Robbie Stevens, and Holger Babinsky. Flexible leading edge flap on an impulsively started flat plate at low reynolds number. In *42nd AIAA Fluid Dynamics Conference and Exhibit*, page 2840.

[24] ML Perry and TJ Mueller. Leading-and trailing-edge flaps on a low reynolds number airfoil. *Journal of aircraft*, 24(9):653–659, 1987.

[25] Kevin J Drost, Heather Johnson, Sourabh V Apte, and James A Liburdy. Low reynolds number flow dynamics of a thin airfoil with an actuated leading edge. In *41st AIAA Fluid Dynamics Conference and Exhibit*, page 3904, 2011.

[26] Charles P Ellington, Coen Van Den Berg, Alexander P Willmott, and Adrian LR Thomas. Leading-edge vortices in insect flight. *Nature*, 384(6610):626, 1996.

[27] Chengjie Wang and Jeff D Eldredge. Low-order phenomenological modeling of leading-edge vortex formation. *Theoretical and Computational Fluid Dynamics*, 27(5):577–598, 2013.

[28] Peter M Mancini, Anya R Jones, Kenneth Granlund, and Michael Ol. Parameter studies on translating rigid and flexible wings. In *52nd Aerospace Sciences Meeting, AIAA*, volume 73, 2014.

[29] PRRJ Stevens, CW Pitt Ford, and Holger Babinsky. Experimental studies of an accelerating, pitching, flat plate at low reynolds number. In *51st AIAA Aerospace Sciences Meeting*, pages 07–10, 2013.

[30] Kunihiko Taira and TIM Colonius. Three-dimensional flows around low-aspect-ratio flat-plate wings at low reynolds numbers. *Journal of Fluid Mechanics*, 623:187–207, 2009.

[31] Wei Shyy, Yongsheng Lian, Jian Tang, Dragos Viieru, and Hao Liu. *Aerodynamics of low Reynolds number flyers*, volume 22. Cambridge University Press, 2007.

[32] Adam Jaroszewicz, Jerzy Stachów, Krzysztof Sibilski, and Andrzej Zyluk. Water tunnel experimental studies of leading edge vortex control on delta wing mav. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 1158, 2011.

[33] Daniel R Morse and James A Liburdy. Vortex dynamics and shedding of a low aspect ratio, flat wing at low reynolds numbers and high angles of attack. *Journal of Fluids Engineering*, 131(5):051202, 2009.

[34] Huai-Te Yu. *Unsteady Aerodynamics of Pitching Flat Plate Wings*. PhD thesis, Air Force Research laboratory, 2014.

[35] Kenneth Granlund, MV Ol, and Luis Bernal. Experiments on pitching plates: force and flowfield measurements at low reynolds numbers. *AIAA Paper*, 872:2011, 2011.

[36] PRRJ Stevens and Holger Babinsky. Low reynolds number experimental studies on flat plates. In *52. AIAA Aerospace Sciences Meeting*, 2014.

[37] EJ Jumper, SJ Schreck, and RL Dimmick. Lift-curve characteristics for an airfoil pitching at constant rate. *Journal of aircraft*, 24(10):680–687, 1987.

[38] Alain Pelletier and Thomas J Mueller. Low reynolds number aerodynamics of low-aspect-ratio, thin/flat/cambered-plate wings. *Journal of Aircraft*, 37(5):825–832, 2000.

[39] Thomas J Mueller. Aerodynamic measurements at low raynolds numbers for fixed wing micro-air vehicles. Technical report, DTIC Document, 2000.

[40] Alexander Radi and Hermann F Fasel. Experimental investigation of laminar separation bubbles on a flat plate. In *40th Fluid Dynamics Conference and Exhibit*, page 4482, 2010.

[41] Michael Gaster. *The structure and behaviour of laminar separation bubbles*. Citeseer, 1969.

[42] Harry Philip Horton. *Laminar separation bubbles in two and three dimensional incompressible flow.* PhD thesis, 1968.

[43] Michael V Ol, Albert Medina, Peter Mancini, and Anya Jones. Revisiting conventional flaps at high deflection-rate. 2016.

[44] JM Walker, HE Helin, and JH Strickland. An experimental investigation of an airfoil undergoing large-amplitude pitching motions. *Aiaa Journal*, 23(8):1141–1142, 1985.

[45] Yongsheng Lian. Parametric study of a pitching flat plate at low reynolds numbers. *Computer Modeling in Engineering & Sciences(CMES)*, 72(1):1–16, 2011.

Aerodynamic Design of a Martian Micro Air Vehicle

T. Désert *, J.M. Moschetta, and H. Bézard
ONERA and ISAE-SUPAERO

ABSTRACT

The present study aims at developing a reliable propulsion system for a rotary wing micro air vehicle (MAV) associated to rovers in order to enhance Martian exploration rate. The main challenge encountered for MAV design is Martian atmosphere's density and speed of sound that are significantly lower than on Earth. Leading to compressible ultra low Reynolds number ($2,000 < Re_{dom} < 10,000$) flows met at blades tip that are unusual and unknown in the biosphere. Consequently, evaluations of numerical tools have been carried out recreating a depressurized experiment. 2D and 3D steady and unsteady Navier-Stokes computations are compared to XFoil for flow behavior apprehension and solver assessment. Based on XFoil's performances evaluations, camber line and thickness distribution have been optimized for 2D incompressible and compressible flows. Optimal shape for a steady solver is a highly cambered airfoil shifting the boundary layer separation downstream. 2D unsteady Navier-Stokes computations show that airfoils delaying heavy unsteadiness generation are producing higher lift and lower drag in 2D than the picked airfoils enhancing vortex production, such as dragonfly airfoils. The impact of airfoil shape on 3D flows is evaluated with a first of its kind experimental campaign in collaboration with CNES. The experimental facility is a ONERA's $18m^3$ tank recreating Martian atmosphere in terms on kinematic viscosity and composition. The tank size allows to reduce wall effects and provide - as compared to previous studies - a more accurate evaluation of rotor performances.

1 INTRODUCTION

Since 2004, three exploration rovers have successfully landed on the Martian surface. Yet, only about sixty kilometers have been explored on 21,000 km of the planet's circumferential path. The slow exploration rate is mainly due to a lack of visibility on the ground. A rotary wing micro air vehicle (MAV) associated to rovers could significantly increase their mobility by providing an aerial point of view of their upcoming pathway. However, Martian atmosphere is far from MAV friendly: density is hundred times lower than on Earth and speed of sound is also lower due to low temperature and different atmosphere composition (96% of CO_2). Hence, a new aerodynamic domain is explored: compressible ultra low Reynolds number flows - Table 1.

	Earth	**Mars**
Density (kg/m^3)	1.225	0.014
Dynamic viscosity (mPa.s)	0.0181	0.0106
Average temperature ($^\circ$C)	15	-63
Speed of sound (m/s)	340	238

Table 1: Atmospheric conditions of Mars and the Earth at ground level

So far, the very few papers operating in this Reynolds number range are mostly studying the impact of Reynolds number [1] and turbulence rate [2] on flow laminarity or evaluating aerodynamic performances of typical airfoils [3] and planform distributions [4]. Only two studies provide airfoil camber line optimization at $Re_c = 6,000$ [5] and $6,000 < Re_c < 16,000$ [6] for incompressible flows. Neither thickness distribution nor compressible optimization have been undertaken yet. Incompressible experimental devices are usually low speed wind tunnels [1], water tunnels [2][3] or nanorotors [5][6]. As far as we know, only two depressurized experiments recreate compressible ultra-low Reynolds number conditions for airfoil [7] or rotor [8] performances measurement. However, neither studies provide a validated computational tool for flow simulation. In the present study, a computational tool is validated on compressible ultra low Reynolds number flows. Moreover, incompressible and compressible airfoil optimizations of both camber line and thickness distributions are carried out. The optimized airfoils are evaluated and compared to airfoils from literature with the validated solver and an experimental campaign.
This paper presents an aerodynamic design of a MAV operating in Martian atmosphere. First, MAV's flight conditions domain is defined. In this domain, steady and unsteady numerical tools are evaluated and compared to *Mars Wind Tunnel* experiment [7]. Then, a compressible airfoil shape optimization process based on 2D steady performances is characterized and carried out step by step for flow behavior apprehension. Finally, the impact of airfoil shape on rotor performances is evaluated thanks to a first of its kind experimental campaign recreating Martian atmospheric conditions and gas composition in an $18m^3$ tank.

*Email address: thibault.desert@onera.fr

2 Description of Martian Micro Air Vehicle flight conditions

Aerodynamic design's domain of the rotary wing MAV correspond to the MAV's flight conditions in hover: Reynolds Number and Mach number range. These two dimensionless quantities depend on atmospheric conditions, MAV's size, weight and rotational speed. The constraints are the diameter of the MAV around 30 cm, its weight of about 200 g and the maximum rotational speed set to avoid locally supersonic flow.

2D laminar steady Navier-Stokes computations set maximal subsonic rotational speed to $12,000$ rpm corresponding to $M = 0.8$ at blade's tip for $c = 238m.s^{-1}$ ($T = -63°C$). Chord Reynolds number range is determined via BEMT evaluations with a number of blades range $n_{blades} \in [\![2, 5]\!]$: $2,000 < Re_{dom} < 10,000$. The upper boundary of the Reynolds number range is enhanced in case of heavier MAV design. The reference Reynolds number is: $Re_{ref} = 3,000$.

3 Assessment of 2D and 3D numerical tools for compressible ultra-Low Reynolds number flows

Numerical tools are not validated on compressible ultra-Low Reynolds number flows yet. Solvers need to be evaluated and compared to an experiment recreating Martian atmospheric conditions : the *Mars Wind Tunnel* [7]. The unsteady solver, elsA [9], is validated on a 3D simulation recreating the test section. The steady quick solver, XFoil, is assessed for a purpose of airfoil optimization process.

3.1 Tohoku's University Mars Wind Tunnel experiment [7]

The *Mars Wind Tunnel* (MWT) experimental device is located at Tohoku's University - Figure 1. It consists of a wind tunnel reproducing Martian atmospheric composition: density and gas. We evaluated the case of the triangular airfoil [7]. This airfoil's strong leading edge camber and sharpness causes significant unsteadiness in the flow, making it interesting for numerical validation. Low density allows to reach

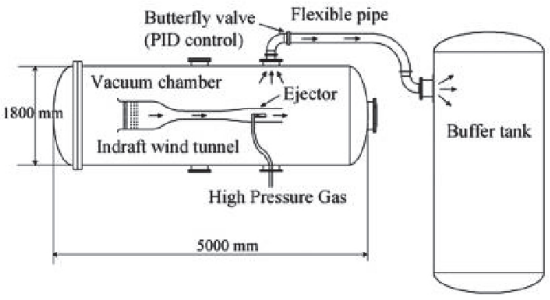


Figure 1: Tohoku's University *Mars Wind Tunnel* experiment

$Re_c = 3,000$ for different Mach numbers - $M = 0.15$ & 0.5. Forces are measured with a balance and pressure distributions are available on the upper part of the airfoil thanks to pressure sensitive paint (PSP). Even if the MWT experiment is supposed to recreate 2D flows, PSP measurement showed a three dimensionality over the airfoil. Moreover, author's 3D LES computations [7] provide greater lift and drag predictions than the experiment while not taking into account the test sections - Figure 2. This phenomenon is due to wall effect on the sides on the test section: the entire experiment needs to be simulated for proper flow and forces evaluations.

3.2 3D unsteady Navier-Stokes solver elsA

The solver elsA was developed by ONERA in the 90's. It is based on an integral form of the compressible Navier-Stokes equations [9]. Considering the Reynolds number range, the hypothesis of laminar flow in a non turbulent environment is conceivable. Transition criteria, such as AHD and Moore, have preliminary confirmed that no transition would occur neither by amplification of Tollmien-Schlichting waves nor by separation bubble. Mesh convergence has been studied for all Navier-Stokes simulations.

3D unsteady simulations recreating the MWT test section are needed in order to recreate the same test conditions as in the experiment. Representative flow conditions ($Re_c = 3,000$, $M = 0.5$) are chosen. $117,000$ cells 2D H-topology meshes with 242 nodes mapping the airfoil's upper surface and 151 for the lower surface have been made taking into account the test section walls with different angles of attack of the airfoil: $5°$, $10°$ and $15°$. The first cell height is set to 2.10^{-4} for chord-normalized coordinate system. From those 2D meshes, 15 millions cells 3D meshes have been constructed with 131 nodes in spanwise direction for a width of 3.3c. As presented

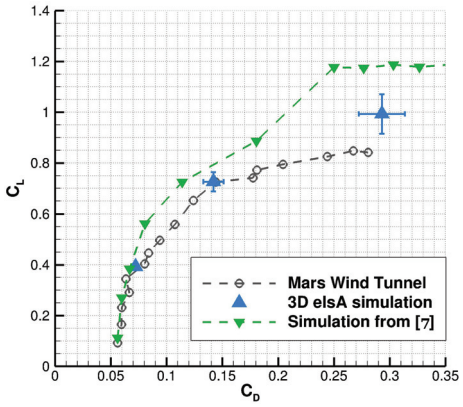


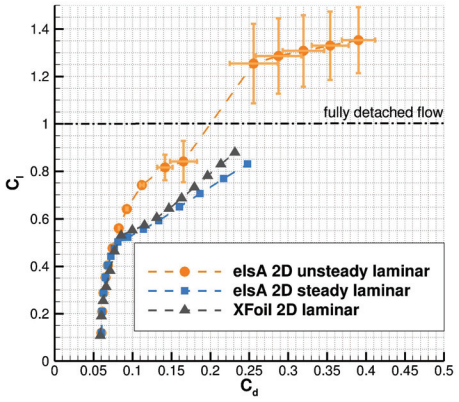
Figure 2: Polar comparing MWT, 3D computations from [7] and 3D elsA simulation (with standard deviation) ($Re_c = 3,000$, $M = 0.5$)

in Figure 2, laminar unsteady Navier-Stokes solver accurately predicts the 3D forces generated at the two first angles of attack ($5°$ and $10°$). However, for the third angle of attack, corresponding to a fully detached flow, computed lift is over-

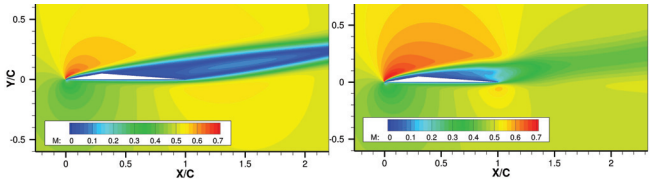
estimated. Since we do not intend to evaluate highly perturbed environment, we consider the solver validated for 3D flows and by extension, we assume that it provides reliable 2D flow predictions.

3.3 2D steady solver assessement for airfoil optimization process

In an airfoil optimization process, a quick and effective tool is needed for performances comparison: unsteady Navier-Stokes simulations' computational cost is too significant. XFoil, a potential flow solver strongly interacted with integral boundary layer formulation, provides a quick steady evaluation of lift, drag and boundary layer state. In order to evaluate laminar XFoil forces prediction, we compare it to laminar steady and unsteady elsA solver on the triangular airfoil test case ($Re_c = 3,000$, $M = 0.5$). Mesh used for these elsA evaluations counts $89,200$ cells with 242 nodes mapping the upper surface and 151 for the lower surface. It is noted that XFoil provides a Karman-Tsien compressibility correction for C_P and external velocity u_e. The integral boundary layer formulation is already valid for compressible flow, therefore, it may be considered as a subcritical compressible solver. As we observe in Figure 3(a), taking into account



(a) Polar comparing elsA steady and unsteady 2D simulations (with standard deviation) and XFoil ($Re_c = 3,000$, $M = 0.5$)



(b) Mach fields in steady case (left) and averaged unsteady (right) ($Re_c = 3,000$, $M = 0.5$, $\alpha = 9°$)

Figure 3: Comparison of different 2D solvers on compressible ultra-low Reynolds number flows

unsteadiness in 2D compressible ultra-low Reynolds number simulations increases the forces generated by the airfoil. This is due to the recirculation zone created by boundary layer separation. For $C_l < 1$, detachment is triggered by 0.3c upper surface discontinuity - Figure 3(b). For $C_l > 1$, flow is fully detached from leading to trailing edge creating a lift and drag overshoot going along with strong unsteadiness. In steady computations using XFoil and elsA, boundary layer separation creates a massive drag rise without significant lift gain causing poor aerodynamic performances. However, forces predictions anterior to heavy boundary layer separation provide valuable hint on airfoil's performances. Therefore, we consider that we can use the steady solver XFoil for airfoil comparison in an optimization process acknowledging that it does not take on board the entire physic of the flow.

4 Airfoil optimization process based on XFoil's performances evaluations

In a purpose of optimization, automatic airfoil generation with finite parameters is primordial. CST method [10] has been chosen because of its ability to recreate any $\mathcal{C}^2$ continuous airfoil shapes with a restricted number of parameters. In this study, we used five parameters to reflect the shape of one curve: camber or thickness distribution. Note that discontinuous airfoils, for example corrugated airfoils, cannot be represented with this parametrization method. However, XFoil is not able to simulate a recirculation zone in a corrugation. The major issue raised by XFoil is convergence: many phenomenon can cause a non-convergence compelling the optimization process to be extremely robust to it. The optimization process consists of evaluating the entire parameters domain with an increasing proximity between the different sets of parameters. As presented in Figure 4, from each input generation N, ten sets of parameters are selected as the most effective airfoils by a performance function. The sets of parameters outside of the zone created by the ten selected are eliminated from the optimization domain: a new domain is formed and a new input generation $N + 1$ is evaluated. Since Martian atmosphere is highly turbulent, the blade's flight conditions can fluctuate. Hence, performance function needs to evaluate the airfoil's competitiveness in the entire hover range of a Martian micro air vehicle. Each set of parameters, representing an airfoil shape, is evaluated on three Reynolds numbers representative of the entire Martian MAV flight domain: $Re_c \in (2,000; 6,000; 10,000)$. The performance function is built as the mean value between lift-to-drag ratio and endurance coefficient over the range of Reynolds numbers and a range of angles of incidence. The process is robust to non-convergence because it does not learn from previous evaluations: we observe an unconverged set of parameters from generation N represented with a green triangle - Figure 4. Since its set of parameters is part of the zone formed by the most effective airfoils of its generation, it is still part of the new generation domain and a close set of parameters is going to be evaluated and converged. However, sufficient proximity on the domain is needed in order to avoid the elimination of

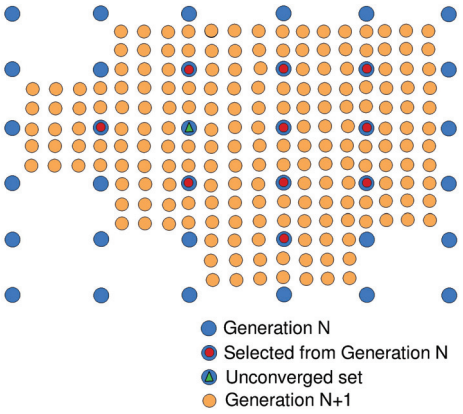


Figure 4: 2D representation of selection process robust to non-convergence

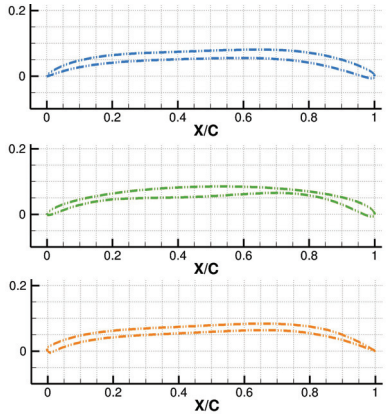
unconverged effective sets of parameters: it demands a very important number of sets and a parallelized process. Each generation counts at least $50,000$ sets of parameters evaluated several times for each of the three Reynolds numbers. Approximately half a million XFoil evaluations are carried out in each generation.

5 Results of compressible and incompressible airfoil optimization for ultra-low Reynolds number flows

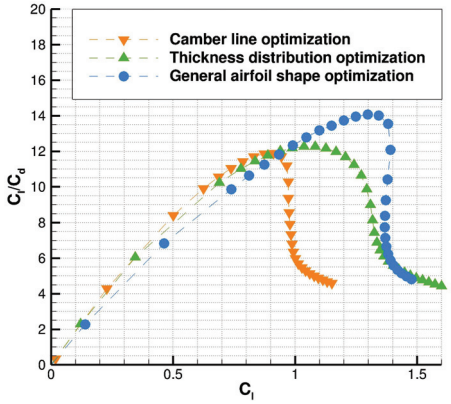
We acknowledge that heavy boundary layer separations are causing poor aerodynamic performances or non-convergence in XFoil. Hence, airfoils enhancing vortex production cannot be truly evaluated and the optimization process aims at generating airfoils shifting the boundary layer separation downstream. For flow behavior apprehension, optimization has been carried out step by step, adding gradual complexity in incompressible flows and then assessing the impact of compressibility.

5.1 Incompressible airfoil shape optimization: camber line and thickness distribution

In the study, three incompressible optimization process have been carried out. Camber line optimization with a prescribed 2% relative thickness distribution having round leading edge and sharp trailing edge. Then, thickness distribution optimization from optimal camber line with a minimum relative thickness constraint set to $T_{min} = 1\%$. Finally, since camber line and thickness distribution are co-dependent, general airfoil shape optimization. As presented in Figure 5(a), camber line shape of each step's optimal airfoils follows the same pattern: a cambered leading edge for proper flow adaptation avoiding early boundary layer separation. A slightly tilted flat middle shape permitting to delay boundary layer detachment. And the highly cambered trailing edge fixes the separation point. Hence, optimal airfoils are producing sufficient lift with high camber while shifting the boundary layer separation downstream. The two incompressible camber line optimizations from literature provide the same general camber line shape for similar range of Reynolds number [5][6]. Optimized thickness distributions show a thin leading edge for proper flow adaptation and a surprising round trailing edge: its impact is evaluated in section 6.2. The optimized thickness distribution with a pre-optimized camber line displays thinner parts at $x/c = 0.15$ & 0.8 that are no longer present with the general shape optimization. Thickness distribution optimization is compensating for the over-cambered line designed for the prescribed leading edge and trailing edge distributions. Figure 5(b) exhibit a better lift-to-drag ratio on a wider range of coefficient of lift at each optimization complexity increment. This is due to better leading edge flow



(a) Incompressible XFoil optimal airfoil shapes: camber line optimization with a predefined thickness distribution (yellow), thickness distribution optimization (green) and general shape optimization (blue)



(b) XFoil's lift-to-drag ratio in flight conditions ($Re_c = 6,000$, $M = 0.1$)

Figure 5: Incompressible general airfoil optimization performances comparison with the different optimization steps

adaptation and trailing edge decompression permitting wider pressure distributions.

5.2 *Compressible airfoil shape optimization: camber line and thickness distribution*

The impact of compressibility has been evaluated for the general airfoil shape optimization. We remind that XFoil may be considered as a compressible solver for subsonic flows: flow rate is set to $M = 0.5$ for avoiding local shocks appearance. As presented in Figure 6, compressible airfoil optimization results in an equivalent thickness distribution with a reduced camber compared to incompressible optimization. In compressible flows, boundary layer separation is more easily triggered: the reduced camber permits to delay high drag production at higher lift, hence, a better lift-to-drag ratio is achieved.

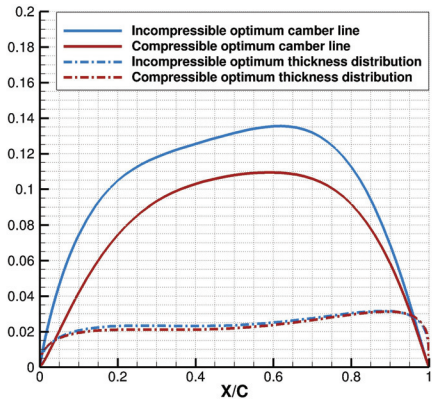


Figure 6: General airfoil shape comparison for a compressible and incompressible optimization

6 Evaluation of the optimized airfoils compared to airfoils picked from literature in 2D unsteady compressible Navier-Stokes simulations

Since the optimization process is based on a simplified steady solver performances evaluations, it is essential to numerically audit the optimal airfoils with a validated solver.

6.1 *Assessment of the optimized airfoils in compressible ultra-low Reynolds number flows*

Figure 7 exhibits the aerodynamic performances of each optimization steps' optimal airfoil in compressible ultra low Reynolds number flows. They have very close unsteady Navier-Stokes lift-to-drag ratio and tendencies are the same for each Reynolds number of the Martian MAV's range. Compressible optimum generates lower drag for $C_l < 1$ than more highly cambered optimized airfoils. However, more highly cambered airfoils suffer the drag rise at higher lift generation. Therefore, the 2D optimal camber line depends on

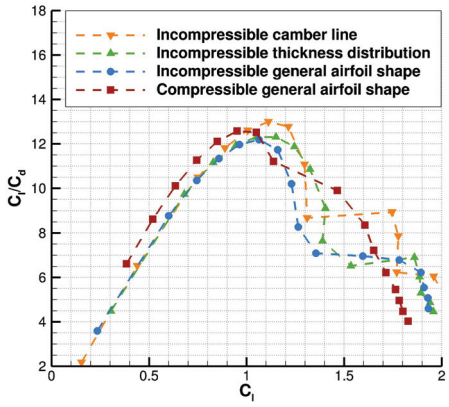


Figure 7: Averaged unsteady 2D N-S lift-to-drag ratio of the optimized airfoils evaluated with elsA in flight conditions ($Re_c = 6,000$, $M = 0.5$)

the aimed range of lift coefficient. We also note that the incompressible camber line optimization provides the best lift-to-drag ratio. It means that thickness distribution optimization for a given camber line did not enhance 2D unsteady aerodynamic performances.

6.2 *Impact of round trailing edge on aerodynamic performances*

The main difference between the optimized and the prescribed thickness distribution comes from trailing edge definition, therefore, we intend to evaluate the impact of this difference by producing an optimal compressible airfoil with a sharp trailing edge replacing its unusual round shape. As presented in Figure 8, gap in 2D aerodynamic performances for different trailing edge thickness distributions is minor. Considering the Reynolds number range, boundary layer growth

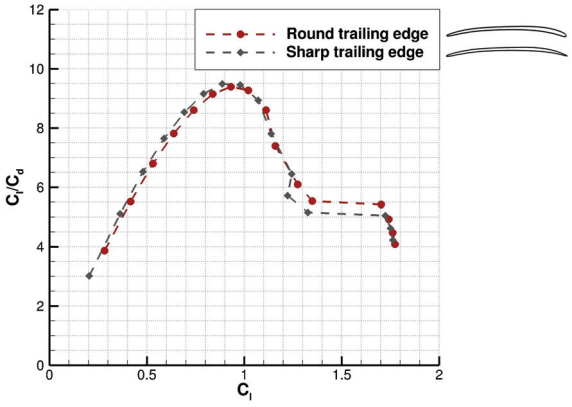
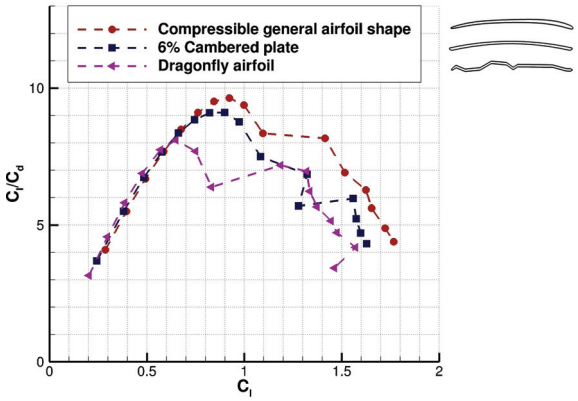


Figure 8: Evaluation of the impact of round trailing edge on aerodynamic performances: averaged unsteady N-S polar in flight conditions ($Re_c = 3,000$, $M = 0.5$)

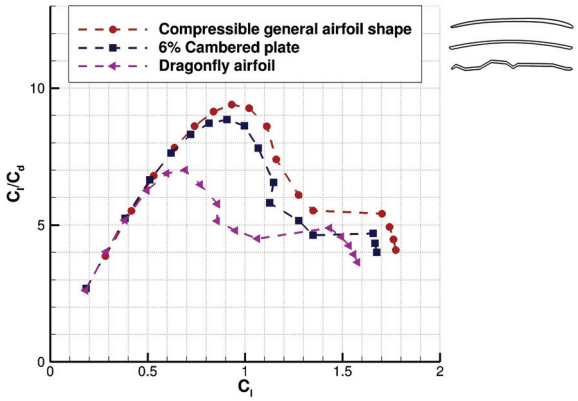
and the recirculation zone triggered by separation at upper surface minimize the magnitude of trailing edge shape. Therefore, in Figure 7, the difference in performances between the incompressible camber line and the incompressible thickness distribution mainly comes from leading edge definition.

6.3 *Compressible and incompressible evaluation of a steady optimized airfoil compared to airfoils picked from literature*

Since the optimized airfoils were build relying on a steady process, we compare the compressible general airfoil shape to airfoils considered in the literature as effective in ultra low Reynolds number flows. The 6% cambered plate has been proven experimentally to be the most effective of the tested airfoil for Martian flight conditions [5][4]. And since biomimicry tends to drive us toward corrugated airfoils enhancing vortex production, a dragonfly airfoil - Figure 10(b) - is picked: it was also experimentally proven to be the most aerodynamically effective from three sections at different radius of a dragonfly wing at ultra-low Reynolds number [11]. Both airfoils are generated with a 2% relative thickness distribution. We observe in Figure 9 that airfoils delaying boundary layer separation provide far better 2D unsteady Navier-Stokes lift-to-drag ratio in ultra-low Reynolds number flows than the dragonfly airfoil. We note that performances are very similar in compressible and incompressible flows for low lift production. At higher lift production, drag rise is more important in compressible flows. Hence, for robustness, design C_l in rotor generation has to be weakened. Figure 10 displays the averaged mach number contours for a production of $C_l \sim 0.85$ of the compressible optimal and the dragonfly airfoil. The corrugated airfoil triggers boundary layer detachment early in chordwise direction degrading 2D performances. However, we acknowledge that only one corrugated airfoil has been computed.
To conclude, XFoil provides valuable hints on airfoil performances permitting to figure out trends in airfoil shapes optimization. 2D unsteady Navier-Stokes evaluations do not show large differences in performances for airfoils postponing boundary layer separation. Nevertheless, a slight 2D performances enhancement could have a greater impact on 3D flows.



(a) Incompressible averaged unsteady N-S polar in flight conditions ($Re_c = 3,000$, $M = 0.1$)



(b) Compressible averaged unsteady N-S polar in flight conditions ($Re_c = 3,000$, $M = 0.5$)

Figure 9: Incompressible and compressible 2D aerodynamic performances evaluated with elsA of an optimized airfoil compared to airfoils from literature [11][5][7]

7 Experimental campaign in a depressurized facility recreating Martian atmospheric conditions

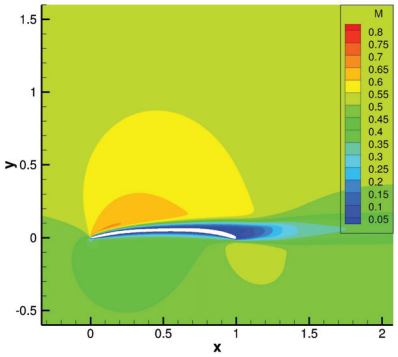
Because flow behaviors might be very different in a 3D rotary system compared to 2D computations, an experimental campaign in a depressurized tank has been carried out in collaboration with CNES. The purposes of the campaign are to understand the flows encountered by experimental rotors and to evaluate the impact of airfoil shape and gas composition on rotor performances.

7.1 *Experimental setup and rotor production*

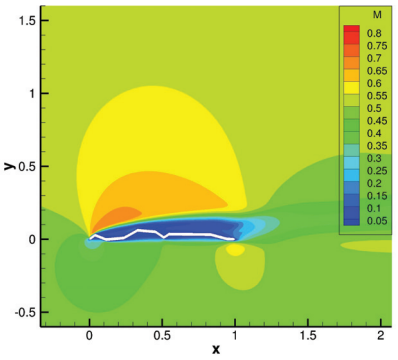
The depressurized facility is a 18 m^3 tank located at ONERA's Fauga center. Inside the tank, an ISAE-SUPAERO testbed is incorporated for thrust and torque measurement - Figure 11. Note that rotor wake heads toward tank's tube, significantly reducing flow recirculation. Martian atmospheric conditions are met in the tank in terms of kinematic viscosity for flight Reynolds number consistency with the two different gases: Air and CO_2 (96%). The aimed pressure inside the tank has been calculated from temperature and gas composition. Hence, performances are compared with dimensionless numbers:

$$C_T = \frac{T}{\rho A(\Omega R)^2} \qquad C_P = \frac{P}{\rho A(\Omega R)^3} \tag{1}$$

Experimental rotors have the same planform distribution from Maryland's experiment [8] - Figure 12. Four airfoil shapes



(a) Averaged unsteady N-S mach fields in flight conditions ($Re_c = 3,000$, $M = 0.5$, $\alpha = 5°$) for the compressible general airfoil shape



(b) Averaged unsteady N-S mach fields in flight conditions ($Re_c = 3,000$, $M = 0.5$, $\alpha = 7°$) for the dragonfly airfoil

Figure 10: 2D averaged unsteady N-S mach fields evaluated with elsA



(a) ISAE-SUPAERO's experimental testbed in the depressurised tank



(b) 18 m^3 depressurized tank located at ONERA's Fauga center

Figure 11: Experimental facility evaluating rotor performances in Martian atmosphere

are evaluated: Maryland's airfoil (6.35% cambered plate), compressible optimum with round and sharp trailing edge and finally the dragonfly airfoil. Rotors and hubs are produced with a liquid resin printer except from the rotor with Maryland's airfoil that was built in carbon.

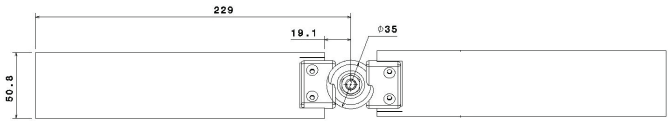


Figure 12: Experimental rotors' planform distribution [8] for evaluating the impact of airfoil shape on rotor performances

7.2 Rotor performances in Martian atmospheric conditions and gas composition

Every experimental rotor has been evaluated in Air and CO_2. Dimensionless performances are comparable in both cases [7] and differences are within the measurement uncertainty. We chose to show the rotor performances in CO_2 since it is closer from Martian atmosphere. Figure 13 displays thrust coefficient of experiments from ONERA and Maryland [8] for different collective pitch angles. Maryland's efforts are under-estimated compared to ONERA's: this is certainly due to flow recirculation inside the smaller tank (0.6 m^3). However, tendencies are comparable thus a correction for compensating the recirculation might be contemplated. For 3D ultra-

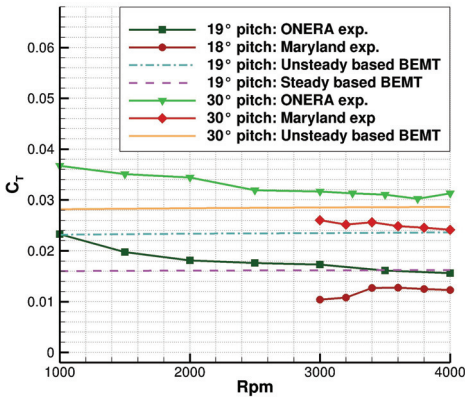


Figure 13: Coefficient of thrust from ONERA and Maryland experiments compared to 2D unsteady and steady based BEMT computations

low Reynolds number flow physics apprehension, Figure 13 also shows expectations from BEMT computations based on 2D unsteady and steady Navier-Stokes polars. For a 19° collective pitch, measured thrust is close to steady based BEMT. However, for a 30° collective pitch, it is closer to unsteady based BEMT than steady based BEMT. Our hypothesis is

that centrifugal forces might tend to stabilize the unsteady flow over the rotor's blades. But, this phenomenon would not be important enough to stabilize a highly unsteady flow at 30° collective pitch. Unsteady CFD rotor computations are in progress for a better understanding of 3D flows encountered by Martian rotors.

The impact of airfoil shape on 3D flows is studied in Figure 14. Rotor performances show the same tendencies as in 2D Navier-Stokes computations for the different airfoil shapes. Highly cambered airfoils shifting boundary layer separation downstream show comparable C_T/C_P and thrust range while the dragonfly airfoil displays poor performances and thrust range. Hence, dragonfly wings are not optimized for gliding at ultra-low Reynolds number.

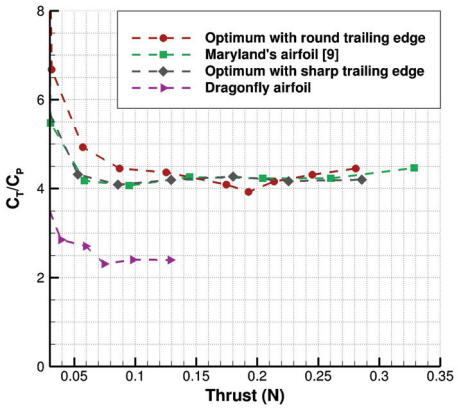


Figure 14: Impact of airfoil shape on rotor performances in CO_2 for the same planform distribution - Figure 12 - and a collective pitch angle of 19° $(1,000 < Rpm < 4,000)$

8 Conclusion

The main conclusions drawn on 2D and 3D compressible ultra-low Reynolds number flows during this numerical and experimental study are the following:

- Laminar unsteady Navier-Stokes solver elsA provides a proper flow simulation and forces prediction in an undisturbed 3D environment.
- 2D steady computations allow to assess the incidence of boundary layer separation and to provide valuable hints on airfoil performances. Making them useful tools for airfoil optimization.
- Effective airfoils for 2D ultra-low Reynolds number flows are highly cambered airfoils with leading edge and trailing edge camber allowing to delay boundary layer separation and unsteadiness production.
- Subcritical compressibility has little impact on 2D aerodynamic performances but eases boundary layer detachment.
- Rotor dimensionless performances are comparable with an Air or CO_2 filled tank for the same flight Reynolds number.
- Tendencies between airfoils performances in experimental campaign recreating Martian flight conditions are the same as with 2D laminar Navier-Stokes computations.

References

[1] R. W. Derksen, M. Agelinchaab, and M. Tachie. Characteristics of the flow over a naca 0012 airfoil at low reynolds numbers. *WIT Transactions on Engineering Sciences*, 2008.

[2] S. Wang, Y. Zhou, and Md. Alam. Effects of reynolds number and turbulent intensity on a low reynolds number airfoil. *AIAA Aviation*, 2014.

[3] S. Sunada, T. Yasuda, and K. Yasuda ans K. Kawachi. Comparison of wing characteristics at an ultralow reynolds number. *Journal of Aircraft*, 2002.

[4] F. Bohorquez. *Rotor Hover Performance and System Design of an Efficient Coaxial Rotary Wing Micro Air Vehicle*. PhD thesis, University of Maryland, 2007.

[5] P.J. Kunz. *Aerodynamics and Design for Ultra-Low Reynolds number flight*. PhD thesis, University of Stanford, 2003.

[6] Z. Liu, L. Dong, J.M. Moschetta, J. Zhao, and G. Yan. Optimization of nano-rotor blade airfoil using controlled elitist nsga-ii. *International Journal of Micro Air Vehicles*, 2014.

[7] P.M. Munday, K. Taira, T. Suwa, D. Numata, and K. Asai. Nonlinear lift on a triangular airfoil in low-reynolds-number compressible flow. *Journal of Aircraft*, May-June 2015.

[8] R. Shrestha, M. Benedict, V. Hrishikeshavan, and I. Chopra. Hover performance of a small-scale helicopter rotor for flying on mars. *Journal of Aircraft*, 2016.

[9] L. Cambier, M. Gazaix, S. Heib, S. Plot, M. Poinot, J.P. Veuillot, J.F. Boussuge, and M. Montagnac. An overview of the multi-purpose elsa flow solver. *Journal Aerospace Lab*, 2011.

[10] B.M. Kulfan and J.E. Bussoletti. "fundamental" parametric geometry representations for aircraft component shapes. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization*, 2006.

[11] A. Kesel. Aerodynamic characteristics of dragonfly wing sections compared with technical aerofoils. *The Journal of Experimental Biology*, 2000.

Study of ducted fans interference for copter type multirotor UAV/RPAS

Stremousov K.*, Arkhipov M.*
* Moscow Institute of Physics and Technology, Department of Aeromechanics and Flight Engineering 16, Gagarina str., 140180, Zhukovsky, Moscow Region, Russia

ABSTRACT

The interference of 17 inch ducted fans with a height of 10% of propellers diameter was studied in hovering regime for the case as it was installed on a quad-rotor copter-type UAV/RPAS. Numerical simulations were provided by solving RANS equations with SST turbulence model using actuator disc with radial distribution of pressure difference according to numerical and experimental investigation of 17 propeller in hovering regime. The straight modeling of the ducted fan with propeller rotation was conducted to obtain higher quality resulting flowfield around the ducted fan. During the 3D numerical simulation of the interference of four ducted fans the improvements of duct geometry were provided in the term of the power consumption with a constant thrust. The wind stability of the quad-rotor copter with four ducted fans was studied by modeling the side wind of different velocities.

1 INTRODUCTION

Since multirotor copter-type vehicles started to be in use for a wide set of applications, such as rescue operations or monitoring for an archeological excavations in mountains (for example Altai in Russia or Alps in Europe), the numerous requirements have been appeared.

The first strong requirement is the hovering time, the longer the flight time, the longer monitoring time is, for example, during the rescue operations iteration. To increase the hovering time the power consumption should be decreased while thrust remains constant. Previous investigations showed that the thrust of a single propeller could be increased by 40% by installing the duct of the same diameter (as single propeller) around the propeller (trimmed along the duct surface) [1]. Thus, the rpm of a propeller could be decreased and the rotor mast pitch/roll moment will be decreased too while the thrust remains constant that would cause the power consumption to drop. The optimal duct also has a height of 60% of the duct diameter that gives a huge side projection as for the copter and strongly affects the second requirement: wind stability. To satisfy wind stability requirements (ability to fly with a wind up to 10 m/s or wind gusts of the same velocity) earlier it was decided to use ducts of 10% of propellers diameter [2]. It was shown that optimal duct of such a height could decrease the power consumption by 20% but the propellers rpm should be slightly increased thus the total power consumption will be decreased by 16% (taking into account interference of the ducted fans on quad-rotor copter) [3].

To study the interference of the ducted fans the 17 inch ducted fan with a height of 10% of diameter obtained earlier during optimization [3] was chosen. The investigation was provided numerically and experimentally. It was decided that the quad-rotor copter to be the experimental vehicle to verify the numerical simulations. Thus the needed thrust was set as 9N which corresponds to the hovering regime.

2 DUCTED FAN INTERFERENCE AERODYNAMICS

The first series of flight tests showed a strange behavior of the copter in the hovering regime. After period since take off the vehicle started to move in roll divergence, such a situation led to a crash if not to land copter immediately. On testing the different types of autopilots, engines, regulators, accumulators and other electronic devices, different propellers the conclusion that this effect to be of duct aerodynamics nature has been done.

To find the reason of such a behavior the numerical simulations were conducted. To provide the numerical simulations on the set of different ducts a structured meshes with H-C topology that contains over 25 million cells and provides resulting Y+ variable of about 0.5-0.75. Duct with actuator disc were situated between two planes of the geometrical symmetry (as it is shown in figure 1) and the boundary condition of rotational periodicity on it. At the actuator disc the pressure change was set as a function of radius, this radial pressure change distribution was taken from the previous modelling of the propeller [4]. The simulations were provided by solving RANS equations with SST turbulence model.

As a result of the simulations the unsteadiness of the thrust was found. The thrust was oscillating with an amplitude of 0.4N while the average magnitude was 9N for all geometries of the duct. Typical dependency of thrust as a function of time is represented in figure 2. The reason of such a behavior is vortices found around the duct. These vortices are starting from the symmetry planes and washing the inner surface of the duct (figures 3 and 4). The vortices provide additional thrust by forming a low pressure zone on the duct surface (figure 5).

Moreover, exchanging the rotational periodicity bound-

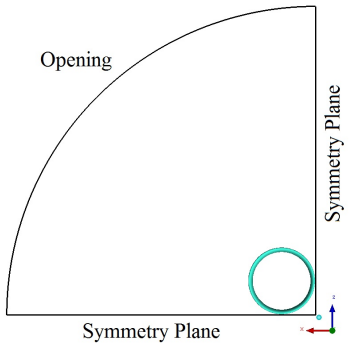


Figure 1: Computational domain for axisymmetric duct.

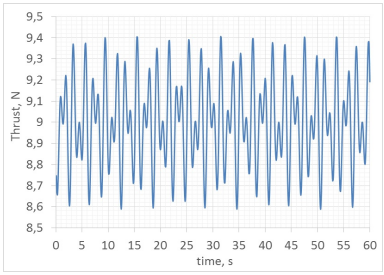


Figure 2: Typical dependency of thrust as a function of time.

ary conditions on the symmetry planes to symmetry boundary conditions it was found that these vortices could oscillate independently along the ducts circle (with an amplitude of 15% of the ducts circle). Thus this instability leads to the thrust and pitch/roll moment oscillations which were observed during the flight tests. The reason of these vortices to be is dawnwash caused by the duct presence. The vortices that occurs on a low set engine intake of the civil aircraft have the same nature.

To confirm that the vortices caused by the duct but not a mistake in the actuator disc the straight modelling of the ducted fan was provided. For that reason, the two-domain structured mesh, which is consisted of 15 million cells, was

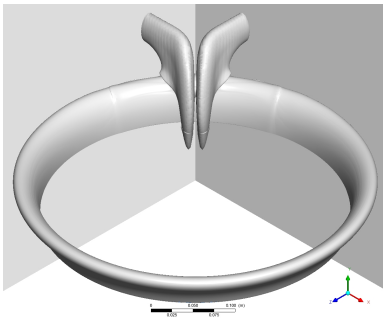


Figure 3: The vortices found. ω = 200 Hz.

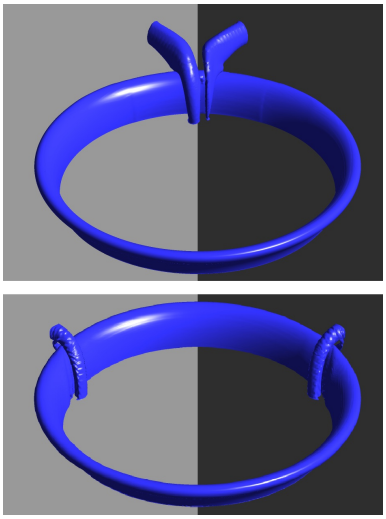


Figure 4: The vortices position at a different time moment. ω = 250 Hz.

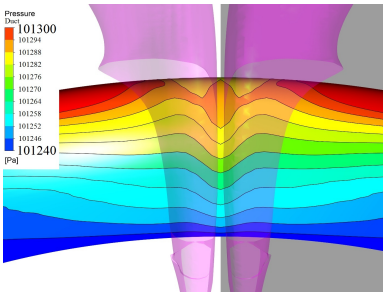


Figure 5: Isopressure lines on the duct surface.

built. First domain around the duct was steady and second one containing propeller rotated around the propeller axis with a rotational speed of 3000 rpm. The disposition of two domains is represented in figure 6. To solve the propeller tip vortices clearly the additional value of cells were set between propeller and the duct, the distance between propellers blade is about 0.5 mm. The cross-section on the quarter-chord of the propellers blade near the duct is shown in figures 7 and 8. In addition, a number of nodes were added along the ducts circle to improve the quality of propellers blade simulation (figure 9). Around the propellers axis the cylinder with a diameter of engine was placed to avoid the zero radii simulation in the solver. The final mesh view is represented in figure **??**.

In straight modelling of the ducted fan the vortices appeared and behave in the same way (figure 10). Each vortex is destructing by the blade while blade is going through the vortex and then appears again. This brings to the high frequency oscillations of thrust on the duct surface in addition to the oscillations mentioned above.

To avoid these vortices oscillation two ways were proposed. First is to change the shape of the duct for the vortices

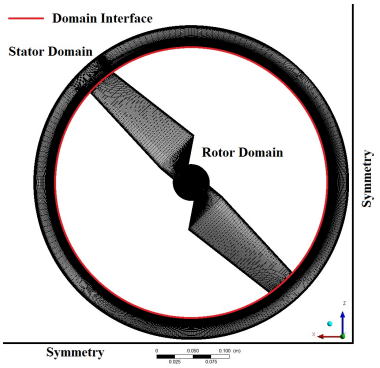


Figure 6: Computational domain for the strict propeller simulation.

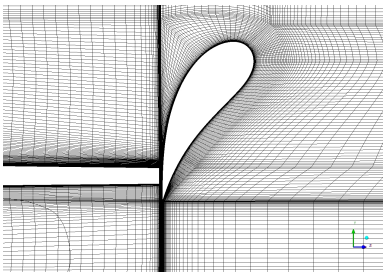


Figure 7: Propelleres balde quarter-chord cross section.

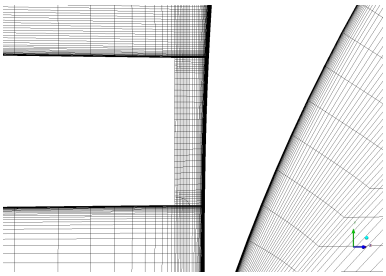


Figure 8: The mesh between duct and blade.

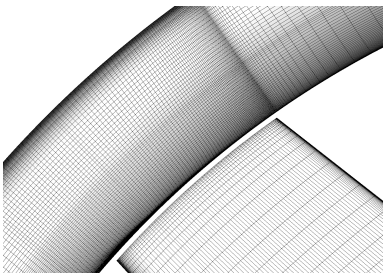


Figure 9: The mesh on the blade, planform view.

to tend to fixed position. Decreasing the incidence of the foil nearest to the symmetry plane, thus it shapes a special cavern for the vortex to be hard to leave it at the flight conditions.

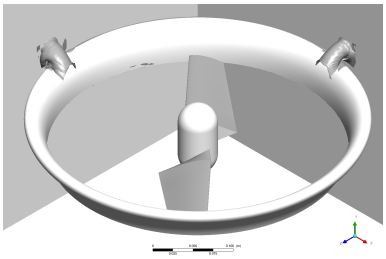


Figure 10: Vortices around the ducted fan. ω = 300 Hz

The numerical simulation of such a duct showed that the amplitude of the vortex oscillation drops up to 0.3% of the ducts circle and 0.1% in terms of the thrust. The typical view of such a duct is shown in figure 11. But the power consumption of such a duct slightly rises in comparison with the axisymmetric duct. It happens because changing the incidence of the airfoil closest to the symmetry the plane in which the propeller rotates become not strictly round and narrowest in the inner space of the duct. To form the propeller/actuator plane strictly round and make it to be narrowest plane it is proposed the foil nearest to the symmetry to move down, the resulting view of the duct is represented in figure 12. The power consumption for this case is 1% lower and is equal to 39.2W, in comparison power consumption of the same propeller without duct in the case when it is installed on the quadrotor copter is 49.6W. The vortices struture around the duct is shown in figure 13.

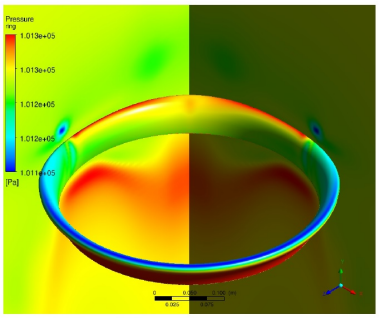


Figure 11: The pressure distribution on the duct with changed incidence of the airfoils closest to the symmetry planes.

The second way to remove the oscillation is to remove the vortices. For this a set of meshes for a numerical simulations were built where the axisymmetric duct was consequently moved away of a copter center with a step of 50% of duct height. The preliminary results of power consumption as a function of the distance between two ducted fans is shown in the figure 14. Now the simulation of the last cases are ongoing and it is expected that power consumption will asymptotically take value of 38.7W which is corresponds to the power consumption of a single axisymmetric ducted fan.

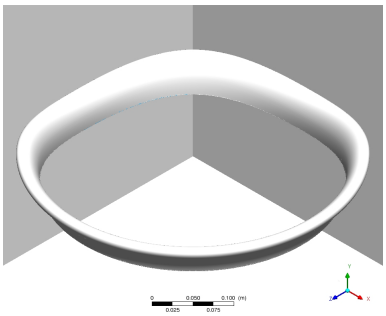


Figure 12: The duct with the airfoils nearest to the symmetry place rotated and moved down.

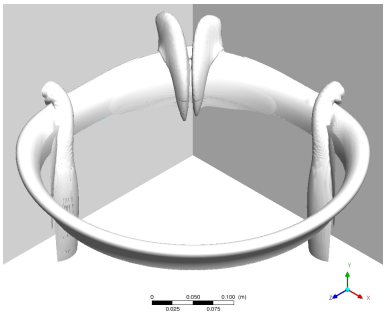


Figure 13: The vortices struture around the duct with changed incidence and moved down airfoils closest to the symmetries.

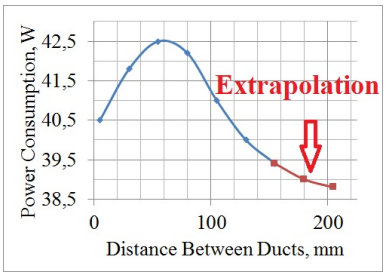


Figure 14: The power consumption dependency from the distance between two ducted fans.

3 Ducted fans interference with a non-zero side wind velocity

To study the ducted fans interference in the conditions of non-zero side wind velocity the mesh on two ducted fans with a symmetry plane was built. Shape of the axisymmetric ducts was taken from the previous study [3]. The H-C topology structured mesh consisted of 50 million cells. Full computational domain is shown in figure 15. The simulations were provided by solving RANS equations with SST turbulence model. The wind velocity of 2.5, 5 and 10 m/s were set on the outer boundaries of the computational domain. In addition, the wind gust of bench and sinusoidal view with a magnitude of 2.5 m/s was also simulated. The actuator discs were set with the same conditions as mentioned above.

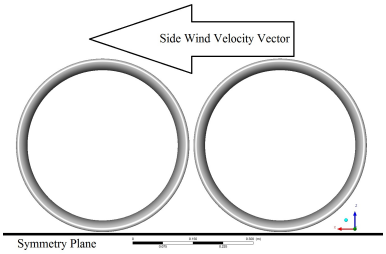


Figure 15: The computational domain for the study of interference in non-zero side wind velocity conditions.

With the appearance of side wind the size of the windward vortex increases while the size of leeward vortex decreases. If the wind velocity grows the effect goes more distinctly. The vorticity of vortices which cores are located along the wind velocity vector decreases as the wind velocity increases until the collapse which is shown in figures 16-18. The location of vortex cores along the actuator changes too. With a growing wind velocity the vortices are breaking down.

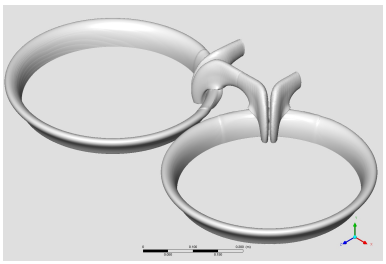


Figure 16: Vortex sheet around duct. Side wind velocity is equal to 0 m/s.

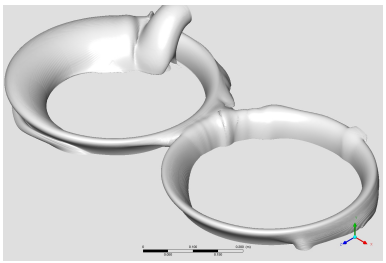


Figure 17: Vortex sheet around duct. Side wind velocity is equal to 2.5 m/s.

With a constant side wind the pitch/roll moment acting on the copter is almost constant; it oscillates in a small range relative to constant value and the cause of this oscillation as described above. The pitch/roll moment dependency from

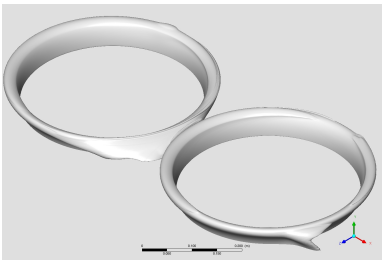


Figure 18: Vortex sheet around duct. Side wind velocity is equal to 5.0 m/s.

time while the wind gust is ongoing is represented in figure 19

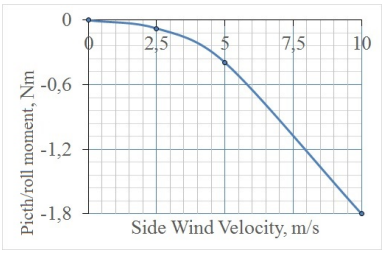


Figure 19: The pitch/roll moment as a function of time.

The transient simulation of working ducted fan confirmed the steady one with 2.5 m/s wind velocity. The form of the bench-like wind gust is given in figure 20. Resultin pitch/roll moment is shown in figure 21. The thrust and the power consumption on both ducted fans obtained from transient simulation is given in figures 22 and 23 respectively.

The modeling was carried out by providing the total two-duct thrust to be 18N while the pitch/roll moment is equal to zero. The power consumption as a dependecy of time in conditions of bench-like side wind gust of 2.5 m/s is represented in figure 25.

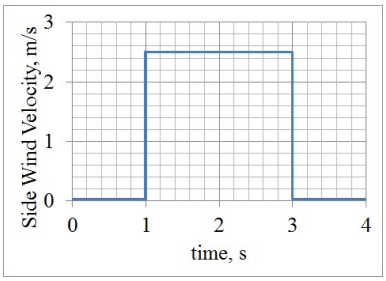


Figure 20: The velocity dependency from time of a bench-like wind gust.

In addition to the numerical simulations provided a simulation of two ducted fans in conditions of side wind gust with a same characteristics as above was carried out but with the

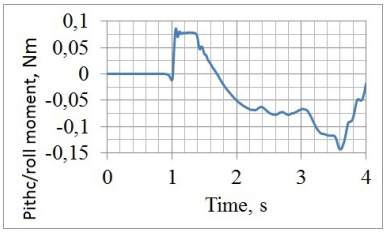


Figure 21: The pitch/roll moment as a function of time.

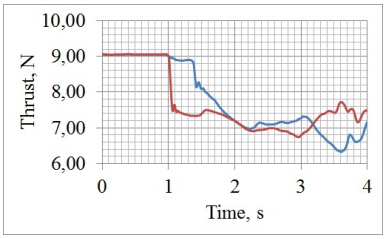


Figure 22: Time dependency of thrust

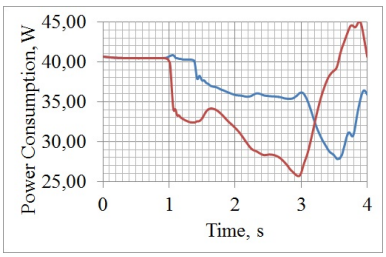


Figure 23: The power consumption as a function of time

regulations of the actuator disc thrust during the simulation. The actuator disc thrust has been regulated in the way that overall thrust of a copter remains constant at the level of 18N and pitch/roll moment is equal to zero. The resulting thrust and power consumption dependencies are shown in figures 24 and 25.

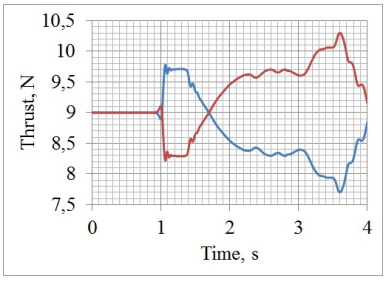


Figure 24: The thrust as a function of time while overall thrust remains constant and pitch/roll moment is equal to zero Nm.

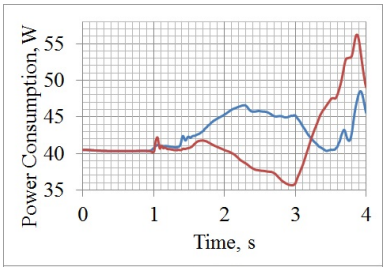


Figure 25: The power consumption as a function of time while overall thrust remains constant and pitch/roll moment is equal to zero Nm.

4 Conclusion

The set 3D numerical simulations were carried out to understand the strange behavior of the copter during the flight tests in hovering regime. The complex vortices structure was found. Each vortex is oscillating along the duct circle providing the thrust to be oscillating too. Thus the nature of pitch/roll moment occured during the copter flight was explained. The straight modelling of a propeller rotating in the duct confirmed the existance of these vortices, moreover during the blade passing the vortex the vortex is breaking down, and the blade thrust is oscillating as the ducts one.

To avoid negative effects caused by these vortices two approaches were developed. The first is to change the duct geometry and the second is to distant ducts from each other. For the first case the vortices were fixed in the same position by rotating and moving down the duct airfoils nearest to the symmetry planes. For the second case it was desided to distant the ducts until the vortices to disappear.

The numerical study of side wind of different velocities and wind gusts of different forms was conducted. The results show that while the velocity of side wind grows the windward vortex vorticity increases and the leeward decreases until both vortices breakdown. The ptich/roll moment, thrust and power consumption as functions of time were obtained.

The thrust and power consumption of windward and leeward ducts, while the overall thrust remains the same and the pithc/roll moment is equal to zero, as a function of time were found.

References

[1] Ostrouhov S.P. *Investigation of the flow pattern near the propeller in the shaped ring and without it at thrust reversal*. TsAGI Science J., Moscow, 2009.

[2] Ageev N.D. Arkhipov M.E. Serokhvostov S.V. Investigation of ducted fans of small height for hexacopter. In *2nd Extremal and Record-Breaking Flights of the RPAS and the Aircraft with Electrical Power Plant*, 2014.

[3] Stremousov K.D. Arkhipov M.E., Serokvhostov S.V. Small height duct design for multicopter fan. In *30th congress of the international councill of aeronautical sciences*, 2016.

[4] Arkhipov M.E. Serokhvostov S.V. Streltsov E.V. Experimental and numerical study of propeller in hover. In *2nd Extremal and Record-Breaking Flights of the RPAS and the Aircraft with Electrical Power Plant*, 2014.

A numerical approach for attitude control of a quadrotor

Huu-Phuc Nguyen, Jérôme De Miras*, Ali Charara and Stéphane Bonnet
Sorbonne universités, Université de Technologie de Compiègne,
CNRS, Heudiasyc UMR 7253, CS 60319, 60203 Compiègne CEDEX

ABSTRACT

This paper deals with a numerical approach to control aggressive maneuvers of a multi-rotor aerial vehicle. The proposed controller uses an approximate tabulated one-step time discretization of the state-space model to find out the outputs of controller. It objective is to minimize the distance between the plant output and a linear well chosen closed loop system used as reference, leading the system to adopt its dynamical behavior. The prediction horizon is only one step time that ensures the execution time is completely bounded. The results from simulation for quadrotor show the performance and robustness of the proposed controller.

1 INTRODUCTION

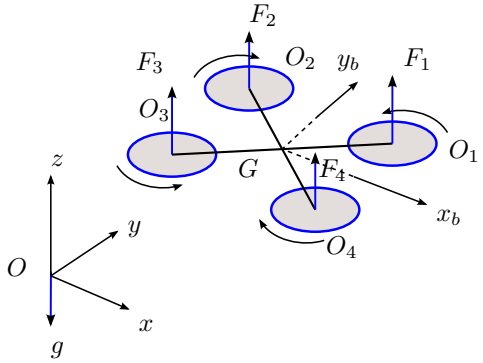


Figure 1: Quadrotor model

A quadrotor is a flying vehicle with four rotors. Figure 1 presents a scheme of the system in a 3D space. The rotor i rotates at speed ω_i and generates a lift force F_i and a drag torque. The quadrotor is an under-actuated system with four inputs and six degrees of freedom. The control problem for the quadrotor is usually divided into two stages: the attitude tracking control and the stabilization of the position. Usually, the position controller generates a desired attitude for the attitude controller. To provide solutions, a lot of control techniques were used: classic PID controllers, adaptive controllers, predictive controllers, controllers based on the Lyapunov criterion, etc.

Looking at the challenge of control of the quadrotor in the case of aggressive maneuvers in the literature, the following works contain many interesting points for studying and comparing. An open loop control was proposed in [1] using reachable sets where the complex aerobatic flights were decomposed into sequences of discrete maneuvers. A trajectory generation for multiple-flips has been proposed in the work of [2] using a simple learning approach. A survey of the methods for attitude control of a rigid body can be found in [3]. The minimal time problem has been considered in the work of [4]. In [5], a quadrotor with flip back behavior was used; the trajectory generation was solved in order to apply a state feedback controller based on a Lyapunov function. A full quaternion based attitude control for quadrotor was introduced in [6]. The authors in [7] designed a controller for flip control on Lie group SE(3). A learning control used in [8] allows the quadrotor doing aggressive maneuvers. The work in [9] have applied an optimal control LQR.

In this context, this paper proposes an attitude control for a quadrotor using a numerical approach. This controller uses a tabulated numerical model to represent the dynamics of the system as a prediction map over one sample time step. Based on this prediction map, the inputs of the system will be calculated upon the admissible input space as a solution of the minimization problem of the difference between the desired output and the predicted output. The rest of the paper will be organized as follows: firstly, the next section deals with the quadrotor dynamics model, then the scheme of the proposed control will be introduced as its application for attitude control of the quadrotor. Finally some results from MATLAB/SIMULINK will be presented.

2 QUADROTOR MODEL

2.1 *Mathematical model*

In the following, vectors and matrices are denoted in bold font. The quadrotor depicted on the Figure 1 is operated by changing the speeds of its four rotors. Using the equation of Newton-Euler, the system is given by the following equations in the body frame $Gx_by_bz_b$:

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= f\mathbf{R}(q)\mathbf{e}_3 - mg\mathbf{e}_3 \\ \dot{q} &= \frac{1}{2}\Xi(q)\Omega \\ \mathbf{J}\dot{\Omega} &= (\mathbf{J}\Omega)\times\Omega + \Gamma \end{cases} \tag{1}$$

with $q = [q_0, q_1, q_2, q_3]^T$ a unit quaternion and

$$\Xi(q) = \begin{bmatrix} -q_1, q_0, -q_3, q_2 \\ -q_2, q_3, q_0, -q_1 \\ -q_3, -q_2, q_1, q_0 \end{bmatrix}^T$$

*Email address(es): (huu-phuc.nguyen,demiras,ali.charara,bonnetst)@hds.utc.fr

The variables and the parameters of the quadrotor are described in the following table:

G	Center of mass
$\mathbf{x} = [x, y, z]^T$	Position of quadrotor
$\mathbf{v} = [V_x, V_y, V_z]^T$	Velocity of the quadrotor
m	Mass of the quadrotor
$\mathbf{J}$	Inertia of the quadrotor
ω_i	i-rotor's speed
ω_{max}	Maximum speed of the rotors
Ω	Angular velocity
g	Gravity
$\mathbf{R}(q)$	Rotation matrix
$\mathbf{e}_3 = [0, 0, 1]^T$	Unit vector
k_t	Thrust coefficient
k_c	Drag torque coefficient
L	Half-distance between F_1F_2

Table 1: Notations

The thrust force and the moments applied on the main body are calculated with the speeds of the rotors as follows:

$$\begin{bmatrix} f \\ \Gamma_x \\ \Gamma_y \\ \Gamma_z \end{bmatrix} = \begin{bmatrix} k_t & k_t & k_t & k_t \\ k_tL & k_tL & -k_tL & -k_tL \\ -k_tL & k_tL & k_tL & -k_tL \\ -k_c & k_c & -k_c & k_c \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{2}$$

The skew-matrix operation of a vector $\mathbf{x} = [x_1, x_2, x_3]^T$ represents an easy way to compute the cross product of two vectors, $\mathbf{x} \times \mathbf{y} = [\mathbf{x}]^\times \mathbf{y}$.

$$[\mathbf{x}]^\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

The inverse operation is denoted as vee-operation such as if $[\mathbf{x}]^\times = \mathbf{X}$ then $\mathbf{X}^\vee = \mathbf{x}$. Considering only the attitude of the quadrotor and the vertical dynamic, the equation above can be written as (3).

$$\begin{cases} \dot{v_z} &= f\mathbf{R}(9) - mg \\ \dot{\mathbf{R}} &= \mathbf{R}[\Omega]^\times \\ \mathbf{J}\dot{\Omega} &= (\mathbf{J}\Omega) \times \Omega + \Gamma \end{cases} \tag{3}$$

where $\mathbf{R}(9)$ is the last item of the rotation matrix $\mathbf{R}$ and $\Gamma = [\Gamma_x, \Gamma_y, \Gamma_z]^T$. The system becomes a full actuated system with four inputs and four degrees of freedom.

2.2 *Full quadrotor model*

There are many toolkits and drag-drop environments to model the dynamics a vehicle using the multibody dynamics theory, for example SimMechanics from Mathworks, MapleSim from MapleSoft. In this paper, we used a quadrotor Parrot-AR2 model built with MapleSim depicted on Figure 2. This model contains a main rigid body, four arms with four brushless motors integrating four propellers to generate the aerodynamics forces. This model also includes some virtual sensors to obtain the quadrotor's state: the position, the velocity, the orientation and the angular velocity. This model is exported to SIMULINK's environment.

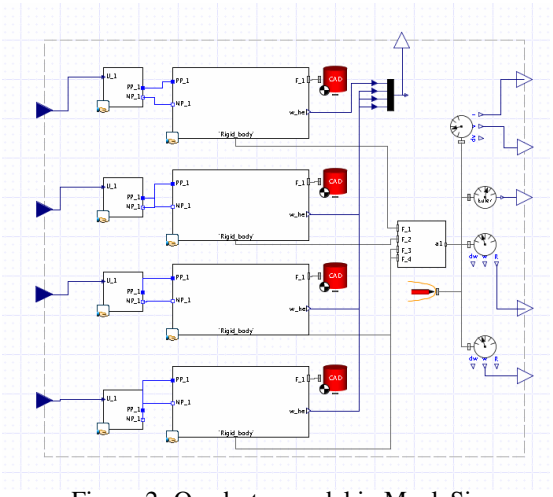


Figure 2: Quadrotor model in MapleSim

The parameters of the quadrotor are given in Table 2.

Parameters	Value	Unit
m	0.506	kg
I_x	2.38×10^{-3}	$kg.m^2$
I_y	3.85×10^{-3}	$kg.m^2$
I_z	5.9×10^{-3}	$kg.m^2$
L	0.15	m
k_t	2.3×10^{-5}	
k_c	2×10^{-6}	
ω_{max}	350	$rad.s^{-1}$

Table 2: Quadrotor parameters

3 PROPOSED CONTROL SCHEME

3.1 *Online control algorithm*

The time-invariant model of of the system has the following form:

$$\begin{cases} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) &= h(\mathbf{x}(t)) \end{cases} \tag{4}$$

where $\mathbf{x} \in \mathcal{S} \subset \mathbb{R}^n$ is the system state vector, $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^d$ its input control vector, and $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^m$ its output vector. The discretized form of the system is needed in order to construct a discrete-time control law. Generally, it is difficult to obtain an exact analytic expression for the discretization of the system. However, the numerical integration of f gives its approximate value for a given state and input vector as the form:

$$\begin{cases} \mathbf{x}_{k+1} &= p_x(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{y}_{k+1} &= p_y(\mathbf{x}_k, \mathbf{u}_k) \end{cases} \tag{5}$$

using a sampling time Δt such as $t_k = k \cdot \Delta t$. Doing the prediction evaluations $(p_x,\ p_y)$ at each needed time by using a numerical integration of (4) would be computationally expensive which is problematic in a real-time implementation. A better approach is to build off-line this prediction in a map using a regular rectangular grid $\mathcal{G}$ constructed on the joint state and input spaces $\mathcal{S} \times \mathcal{U}$. For each point of $\mathcal{G}$, the equation of the system is solved over one sample time step using a simulation tool such as Simulink or using an ordinary differential equation solver such as Runge-Kutta. The solution vectors are then stored in a table (the map). The values of $p_x,\ p_y$ at unknown points can be interpolated from that table by using interpolation technique. The barycentric linear interpolation as described in [10] has been chosen for instance. The linear interpolation leads to errors proportional to the square of the grid size but it presents a low computing effort needed to get the interpolated values.

The aim of the controller is to make the error $\hat{\mathbf{y}} = \mathbf{y} - \mathbf{y}_c$ between the plant output $\mathbf{y}$ and a given set point $\mathbf{y}_c$ to be driven to zero. Defining the state $\hat{\mathbf{x}}^g$ of a homogeneous stable linear system:

$$\dot{\hat{\mathbf{x}}}^g(t) = \mathbf{A} \cdot \hat{\mathbf{x}}^g(t), \hat{\mathbf{x}}^g \in \mathbb{R}^n, \mathbf{A} \in \mathcal{M}^{n \times n}. \quad (6)$$

and $\hat{\mathbf{x}}^g = \mathbf{x}^g - \mathbf{x}_c$ with $\mathbf{x}^g$ an internal state reference, the output $\mathbf{y}^g$ of the corresponding system will converge to the given set point $\mathbf{y}_c$. The Hurwitz matrix $\mathbf{A}$ is selected according to the known dynamics of the system to control. A discrete representation of the internal reference $\mathbf{x}^g_{k+1}$ at step $k+1$ can be computed from (6) at step k as:

$$\mathbf{x}^g_{k+1} = e^{\mathbf{A} \cdot \Delta t}(\mathbf{x}_k - \mathbf{x}_c) + \mathbf{x}_c. \quad (7)$$

Consider now that one of the objective of the control is to reduce as mush as possible the distance between $\mathbf{x}^g$ and $\mathbf{x}$ at each step time, as in a sliding mode, and $\mathbf{x}^g$ converges naturally to $\mathbf{x}_c$ then we can say that $\mathbf{y}$ will converge to $\mathbf{y}_c$.

To construct the algorithm, a vector $\mathbf{v} = \left[\mathbf{x}^T, \mathbf{y}^T\right]^T$ is used. The state vector $\mathbf{x}$ and the output vector $\mathbf{y}$ are then extracted from $\mathbf{v}$:

$$\mathbf{x} = \mathbf{P}_x \cdot \mathbf{v}, \mathbf{y} = \mathbf{P}_y \cdot \mathbf{v}$$

At step $k+1$ this vector is

$$\mathbf{v}_{k+1} = \left[p_x(\mathbf{x}_k, \mathbf{u}_k)^T, p_y(\mathbf{x}_k, \mathbf{u}_k)^T\right]^T = p(\mathbf{x}_k, \mathbf{u}_k). \quad (8)$$

The objective of the controller is to find at each time-step k a control vector $\mathbf{u}_k$ that will make the next-step plant output $\mathbf{y}_{k+1}$ be as close of $\mathbf{y}^g_{k+1}$ as possible:

$$\mathbf{u}_k = \arg\min_{u \in \mathcal{U}} \left\| \mathbf{y}^g_{k+1} - \mathbf{P}_y \cdot p(\mathbf{x}_k, \mathbf{u}) \right\|_2. \quad (9)$$

Due to the interpolation error, the modeling error in (4) and the unknown disturbance, there exists a prediction error between the predicted plant state and the outputs at step k obtained from data at step $k-1$ and their estimations obtained from the measurements at step k. The error signal for the output predictions is handled similarly and the error vector for the combined vector $\mathbf{v}_k$ is defined by:

$$\epsilon_k = p(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) - \mathbf{v}_k.$$

The use of the prediction map to calculate the input value needs compensate this error. The error dynamics is supposed slow relative to the controller. But in the case of noisy conditions, the error signal contains high frequency. Compensating for high frequency errors can indeed lead to oscillations and controller instability. To overcome this phenomena, a low-pass filter is used to remove the high frequency. The error signal for the state prediction used at step k can be written as:

$$\varepsilon_k = \varepsilon_{k-1} + \alpha \epsilon_k \quad (10)$$

with α a damping coefficient in $(0, 1]$, forming a numerical low-pass filter.

The complete algorithm for closed-loop system from [11] is shown in Figure 3.

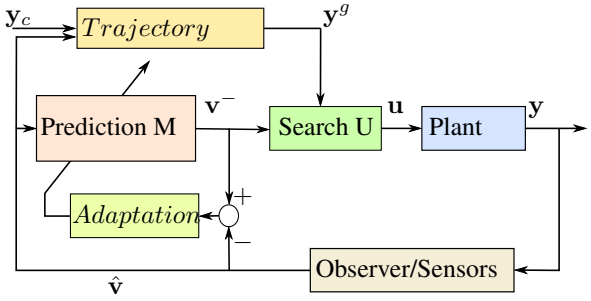


Figure 3: The closed - loop system

At step $k+1$ the output $\mathbf{y}_k$ is measured. Then the block "$Observer/Sensors$" is used to estimate state and output vector and returns $\hat{\mathbf{v}}_k$ an estimation of $\mathbf{v}_k$. After that the error ε_k and the prediction $\mathbf{v}^-_k$ of $\mathbf{v}$ is calculated by the block "$Adaptation$" using (10) and the block "$Prediction\ M$" using (8) respectively, allowing computation of the internal reference $\mathbf{x}^g_{k+2}$ following Eq.(7) by the block "$Trajectory$". Finally, the input vector is calculated using the block "$Search\ U$" that contains the algorithm to find the approximate solution of (9). In a perfect world, the control input $\mathbf{u}_k$ is supposed to be readily computed and immediately applied to the plant from the measurements at step k without any delay of the state estimation. In a real time system, some methods were proposed to compensate for a delay between the output control and the measurement, for example [12].

3.2 Optimization algorithm

As mentioned before, the expression for p is unknown then the minimization problem (9) has to be solved using a

derivative free optimization solver. In this subsection an iterative algorithm will be used to approximate the solution of (9). This algorithms starts by using the input interval $\mathcal{U} = [\underline{\mathbf{u}}, \overline{\mathbf{u}}]$ that is used to build the prediction map.

$$\mathcal{U}_{i\in 1,\ldots,d} = \left\{ u_j | u_j = \underline{u}_i + j\frac{\overline{u}_i - \underline{u}_i}{N_i}, j \in 0,\ldots,N_i \right\}$$

The input space is divided using triangularization technique that is based on the simplex and the affine-envelope of a set of vector. These simplexes are also used at the interpolation step to calculate the barycentric coordinates of a point that is not belong to the rectangular grid $\mathcal{G}$. In this algorithm, the simplexes and the affine-envelope are defined as:

$$\mathbf{conv}\,(\mathcal{X}) = \left\{ \sum_{i=1}^{k} \omega_i \cdot \mathbf{x}_i | \mathbf{x}_i \in \mathcal{X}, \omega_i \in \mathbb{R}^+, \sum_{i=1}^{k} \omega_i = 1 \right\}$$

$$\mathbf{aff}\,(\mathcal{X}) = \left\{ \sum_{i=1}^{k} \omega_i \cdot \mathbf{x}_i | \mathbf{x}_i \in \mathcal{X}, \omega_i \in \mathbb{R}, \sum_{i=1}^{k} \omega_i = 1 \right\}$$

The algorithm based upon the work presented in [11], [13] is illustrated on Figure 4. The idea of this algorithm is to find out a simplex σ_f in the inputs space containing the solution of the problem (9) that minimizes the distance of the corresponding simplex Γ_f in the output space to the reference point $\mathbf{y}^g$. Firstly, one input simplex σ_0 is chosen permitting to calculate its output simplex Γ_0. A quasi-gradient is then built based on the projection of $\mathbf{y}^g$ onto Γ_0. The simplex σ_0 is reflected about its edge satisfying the output simplex moves toward the reference point $\mathbf{y}^g$ while keeping its beside the input space. This procedure is repeated until the final simplex σ_f is found out. Finally, the control input is calculated as an orthogonal projection of this point onto the solution simplex σ_f. To compensate the error of the prediction map ε_k, the output simplex Γ is calculated by $\Gamma = \mathbf{P}_y.p(\sigma, \mathbf{u}) - \varepsilon_k$.

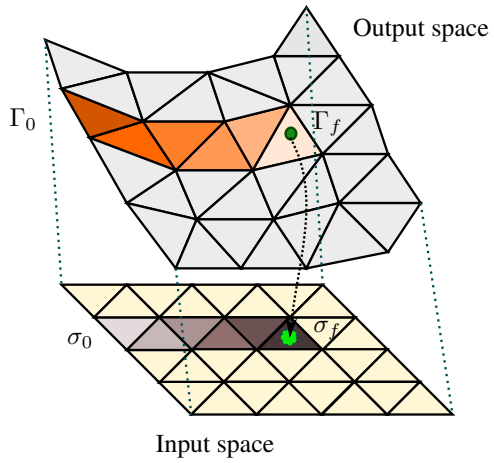


Figure 4: Optimal algorithm visualization

3.3 *Implementation of the algorithm for the quadrotor*

For the implementation of the algorithm, the prediction map should be built from the equation dynamics of the system. The reference equation is (3). The difficulty is that the orientation represented by the unit quaternion contains a constraint. The norm of the quaternion is equal to one $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$, so that, it is not possible to use directly q as four independent elements. In the other hand, the use of the three Euler angles contains a singularity. Avoiding these problems, a new variable will be introduced in order to represent the orientation dynamics. Define now the rotation error between two steps as

$$\Delta\mathbf{R}_{k+1} = \frac{1}{2}\left(\mathbf{R}_{k+1}^T\mathbf{R}_k - \mathbf{R}_k^T\mathbf{R}_{k+1}\right) \quad (11)$$

Note that the rotation error matrix $\Delta\mathbf{R}_{k+1}$ can be verified being a skew-matrix. So that, this matrix is represented by using three independent variables $\mathbf{e}_R^{k+1} = \Delta\mathbf{R}_{k+1}^{\vee}$. Then from (3), the sub-dynamics of the quadrotor can be written in a discrete time domain as (12).

$$[v_z^{k+1}, \mathbf{e}_R^{k+1}, \Omega_{k+1}]^T = F(f_k, \Gamma_k, v_z^k, \mathbf{e}_R^k, \Omega_k, \mathbf{R}_k(9)) \quad (12)$$

Recall that the rotation matrix with the angular velocity Ω_k and the sampling time T_s is calculated by

$$\mathbf{R}_{k+1} = \mathbf{R}_k \exp\left(T_s\Omega_k\right) \quad (13)$$

Thanks to the Rodrigues' rotation formula, $\Delta\mathbf{R}_{k+1}$ has the approximation form:

$$\Delta\mathbf{R}_{k+1} \approx -T_s[\Omega_k]^{\times} \quad (14)$$

That gives the rotation error approximated by

$$\mathbf{e}_R^{k+1} \approx -T_s\Omega_k \quad (15)$$

Combination of (15), (12) and (3) allows a method to build an approximation map of the attitude dynamics of the quadrotor. Once the prediction map is built, the control algorithm above will be used for the stabilization of the attitude of the quadrotor.

4 SIMULATION RESULTS

In this section, we will introduce some simulation results from MATLAB/SIMULINK. The quadrotor model used in these simulations has been exported from MAPLESIM. The sampling time $T_s = 0.01s$ was chosen for the prediction map. A 100 Hz frequency for the control of a quadrotor is rather a low value. The internal reference linear state-space system for each axis $\mathbf{e}_R$, Ω and v_z has an equivalent canonical first-order transfer function $W(s) = \frac{\tau_s}{\tau_s.s+1}$ with a time constant of $\tau_s = 30^{-1}\ s$, $\tau_s = 50^{-1}\ s$ and $\tau_s = 2^{-1}\ s$ respectively. In the first simulation, the attitude trajectory is constructed from three RPY angles (roll, pitch, yaw) $y_{di} = 0.5\sin(t-i)$

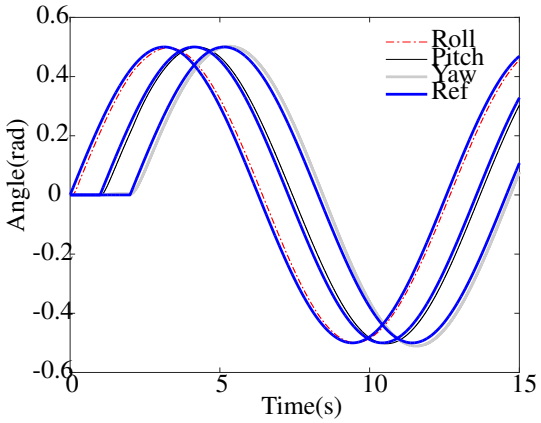


Figure 5: RPY angles and its reference

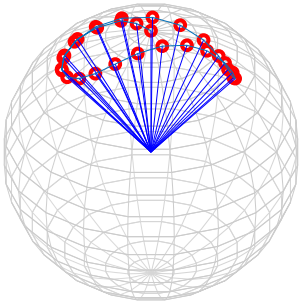


Figure 6: Attitude tracking

with $i = 1, 2, 3$ for roll-pitch-yaw angle respectively. The response angles from the quadrotor and its reference are shown in Figure 5.

This figure shows that the quadrotor tracked well the desired attitude. On the illustration shown in Figure 6, the trajectory generated by the vector $\mathbf{e}_3 = [0, 0, 1]^T$ converges to its reference.

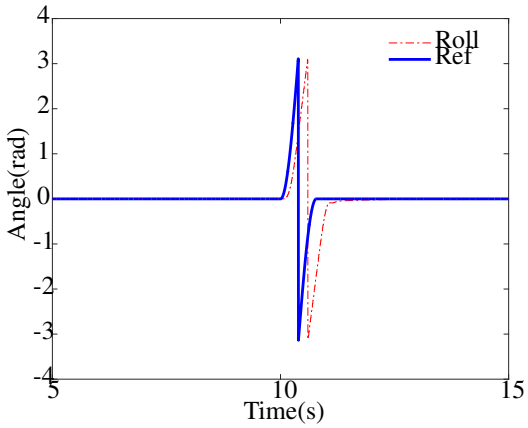


Figure 7: Roll angle and its reference

The second test is performed by doing a single flip about the x-axis. The pitch and yaw angle are driven to zeros while the roll angle was tracking a rotating trajectory. In this test, the desired vertical velocity v_z is controlled so as to keep the position of the quadrotor at 2.5 m. Figure8 shows that the vertical unit vector z_b rotates well one perfect round about x-axis. This significant is also justified in Figure 7.

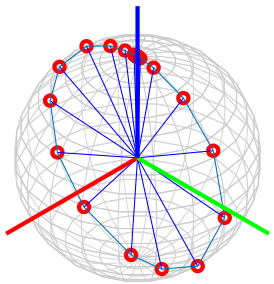


Figure 8: Rotation about x-axis

Figure 9 shows the vertical position z, its reference and the vertical velocity. The position z (dash-dotted curve) tends to its reference 2.5 m. The descent while flipping explains why the user usually accelerate the quadrotor before and during doing flips.

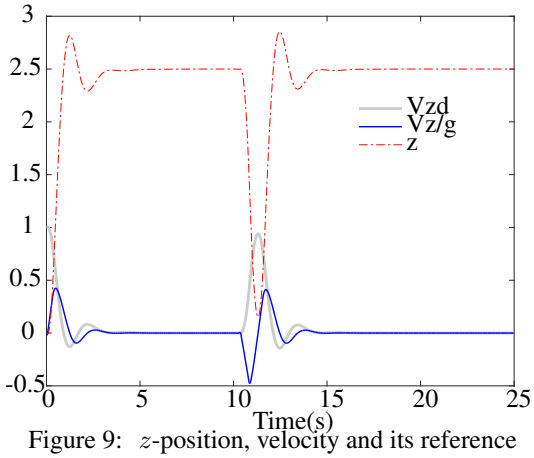


Figure 9: z-position, velocity and its reference

Figure 10 shows the rotors' nominal speed ω_i/ω_{max}. In hover flight, the rotors' speed are kept at a constant value. To do a single flip, Figure 10 shows the left rotors' speed ω_1 and ω_2 are increased while the right rotors' speed ω_3 and ω_4 are decreased in order to generate a torque that twists the quadrotor about x-axis. After the loop is perfectly done, the controller has gotten the smallest speed for the rotors to stabilize faster the roll angle. Finally, the vertical position is compensated.

Figure 11 shows the nominal inputs $T = f/(k_t * \omega_{max}^2)$, $G_1 = \Gamma_x/(k_t * L * \omega_{max}^2)$, $G_2 = \Gamma_y/(k_t * L * \omega_{max}^2)$ et $G_3 = \Gamma_z/(k_c * \omega_{max}^2)$. The nominal thrust T also shows that the acceleration was increased to accelerate the quadrotor in order to perform the looping while the roll torque $G1$ rotates the system about x-axis.

5 Conclusion

This paper presents a numerical tabulated approach to the attitude control problem. The numerical behavior of the sys-

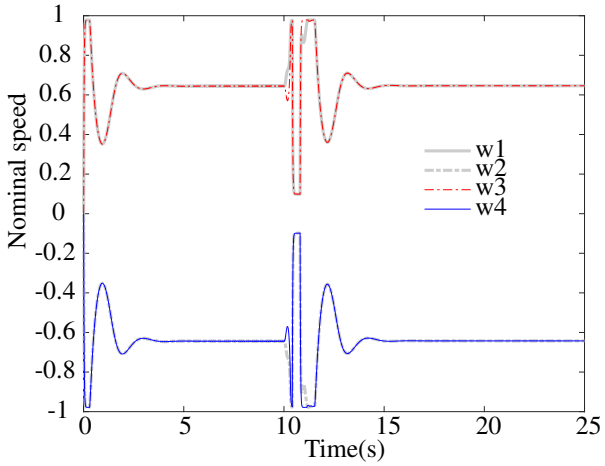


Figure 10: Rotors' nominal speed

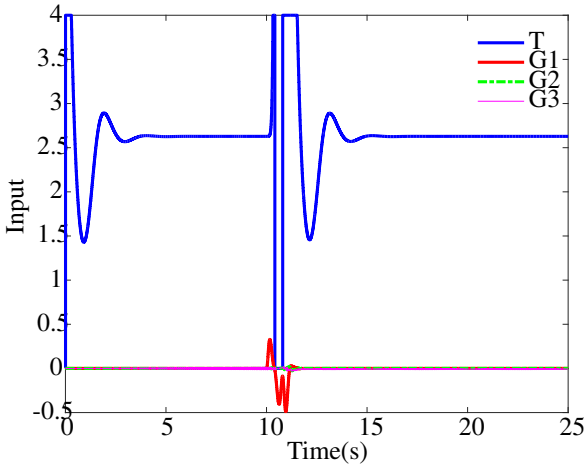


Figure 11: Looping: input

tem is a hybrid model that uses the incremental error between two time steps, avoiding the singularity and ambiguity of the attitude representation. This control could be combined with a position control in order to track complex position trajectory. The validation of the proposed controller on a embedded system requires a good observer for the attitude. Moreover, the prediction map could be improved online using learning-based method.

References

[1] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin. Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1649–1654, 2010.

[2] S. Lupashin, A. Schollig, M. Sherback, and R. D'Andrea. A simple learning strategy for high-speed quadrocopter multi-flips. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1642–1648, 2010.

[3] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch. Rigid-body attitude control. *IEEE Control Systems*, 31(3):30–51, 2011.

[4] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2520–2525, 2011.

[5] Cutler Mark and How Jonathan. *Actuator Constrained Trajectory Generation and Control for Variable-Pitch Quadrotors*. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, 2012.

[6] E. Fresk and G. Nikolakopoulos. Full quaternion based attitude control for a quadrotor. In *Control Conference (ECC), 2013 European*, pages 3864–3869, 2013.

[7] F. Goodarzi, D. Lee, and T. Lee. Geometric nonlinear pid control of a quadrotor uav on se(3). In *Control Conference (ECC), 2013 European*, pages 3845–3850, 2013.

[8] T. Tomix, M. Maier, and S. Haddadin. Learning quadrotor maneuvers from optimal control and generalizing in real-time. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1747–1754, 2014.

[9] Sara Spedicato, Giuseppe Notarstefano, Heinrich H. Bulthoff, and Antonio Franchi. *Aggressive Maneuver Regulation of a Quadrotor UAV*, pages 95–112. Springer International Publishing, Cham, 2016.

[10] R. Munos and A. Moore. Barycentric interpolators for continuous space & time reinforcement learning. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 1024–1030, Cambridge, MA, USA, 1999. MIT Press.

[11] Stéphane Bonnet. *Appoches numérique pour la commande des systèmes dynamiques*. PhD thesis, Université de Technologie de Compiègne, France, 2008.

[12] J. De Miras and S. Bonnet. Nonlinear control of a magnetic levitation shaft by numerical inversion of its behavioral model. In *13th IEEE European Control Conference*, Strasbourg, France, June 2014.

[13] H-P. Nguyen, J. De Miras, S. Bonnet, and A. Charara. Nonlinear control of the pvtol aircraft by numerical inversion of its behavioral model. In *23th IEEE International Conference on Control Application*, Antibe, France, October 2014.

Application of a Switching Control Strategy to Extract Energy from Turbulence by a Fixed-wing UAV

F. Pasquali, Y. Brière *, and N. Gavrilovic
Université Federale de Toulouse - ISAE Supaero, 10 av. Edouard Belin, 31400 Toulouse, France

ABSTRACT

The objective of this paper is to design a control law to allow a small fixed-wing Unmanned Aerial Vehicle to extract energy from atmospheric turbulence. From literature data the properties of atmospheric gusts at low altitude are discussed and a single point measurement is proved to be representative of the wind field. The longitudinal flight dynamics of the aircraft is analyzed and the phugoid mode is found to be the main driver of the energy extraction process. A switching controller that places the poles of the phugoid mode depending on the instantaneous variation of energy of the aircraft is designed. Statistical simulations show an increase of energy of the aircraft when this strategy is applied.

1 INTRODUCTION

A major problem of Unmanned Aerial Vehicles consists in the duration of the battery, which nowadays strongly reduces the endurance of a mission; anyway, the atmosphere containts energy in the form of wind. While the energy in everyday gusts is negligible compared to the one of a civil airplane [1], it represents a large quantity of the energy of small UAVs, whose flight domain is close to the one of a bird, thus making energy extraction worth to be investigated.
Energy harvesting from turbulence is not a topic with well-defined boundaries and results, thus several research domains have to be taken into account. First, the experience of glider pilots [2, 3] and the flight of birds are a source of data. Second, in the '90s, several authors [4, 5, 6] addressed the response of civil planes to gusts (for strong winds due to extraordinary meteorological conditions) in particular in the phases of take-off and landing, leading to the Total Energy Control System (TECS), common on civil aircrafts nowadays. Specifically about energy harvesting from the atmosphere, the literature have dealt with three phenomena: thermals [7] that are widely known by glider pilots, wind shear which is exploited by marine birds via a technique called 'dynamic soaring' [8] and eventually gusts. For the first two cases, an apriori knowledge of the wind field is used, contrary to the stochastic nature of turbulence.
Two approaches are common to deal with atmospheric gusts. One is to investigate the natural capability of the wing to extract energy, in terms of "microlift" force [9], aeroelastic response [10] and overall design [11]. The other is to implement a suitable control law; the most used technique [12, 13, 14, 15] computes a set of feedback gains via optimization of the energy of the aircraft on a set of gusts.

The main contributions of this paper are:

1. The evidence that a single point measurement of the wind field is representative of the energetic turbulent wind, thus just one probe is needed.
2. The computation of the transfer functions from the turbulent wind velocity to the energy of the aircraft and the identification of the phugoid as the main dynamics affecting this process in the case of horizontal gusts.
3. The design of an energy extraction control law that does not rely on numerical optimization but on the switching between two controllers to tune the pulsation of the phugoid dynamics depending on the instantaneous loss or gain of energy by the aircraft (and the proof of its stability).
4. The simulations on Von Karman horizontal gusts to show that the controller is able to extract energy from gusts without a consistent increase of the elevator activity with respect to a just stabilizing controller

The paper is structured as follow. In 'Section 2' the main results and problems of an optimal control technique are presented. 'Section 3' is dedicated to the analysis of low altitude atmospheric turbulence. 'Section 4' deals with flight mechanics in presence of wind gusts. In 'Section 5' we design the control law and eventually, in 'Section 6' we perform simulations on gusts.

2 LITERATURE REVIEW

The main references dealing with energy extraction from wind gusts using an appropriate control law are the works by Langelaan [12] and Patel et al [15]. They use similar methods to optimize the gains of a proportional feedback law on a set of gusts so to obtain the smallest decrease of energy while gliding. Patel [15] focuses on vertical gusts using a point-mass glider model and proves his results by flight tests. He introduces the idea to use a feedback law depending on the energy and obtains a reduction of energy losses of 36% with respect to a stabilizing controller. Langelaan [12] uses the complete longitudinal rigid dynamics, takes into account vertical and horizontal gusts acting at the same time on the aircraft and proves a reduction of energy losses of around 40%.

*Email address(es): contact yves.briere@isae.fr

Even if their results are promising, the usage of an optimal control strategy leads to some unclear points.

1. Langelaan [12] obtains a set of optimal gains that show a large scatter and sometimes changes in sign when the optimization is performed on different sets of gusts.
2. The relations among the characteristics of the aircraft, the properties of turbulence and the control strategy are difficult to be analyzed using numerical methods, so the quantities that have primary importance in the energy extraction process are not identified.
3. When applying the energy extraction control law, a large increase in the control action is required, thus a high amount of power from the batteries is necessary.

3 Atmospheric Turbulence at Low Altitude

3.1 What do we mean by turbulence?

The aim of this section is to describe turbulence so to represent an input for our airplane model. Turbulence, due to its "randomness", can be addressed only in a statistical way: several works in the literature, nicely reviewed in [16], measure the Power Spectral Density (PSD) of the wind. We deal only with the turbulent range (at higher frequencies than the spectral gap identified in [17]), while the low frequency phenomena are considered to create a constant wind; when referring to turbulent wind speed, we consider the oscillations of velocity (having zero mean value) with respect to the constant wind. Only the turbulence in the vertical plane is considered, since we investigate only the longitudinal dynamics of the plane.
Common turbulent spectra such as Von Karman [18] or Dryden depend on four parameters:

1. Variance of the turbulent velocity σ^2: corresponding to the total power of turbulence.
2. Injection length L: a measure of the scale of the energetic turbulent eddies (it represents the autocorrelation length of turbulent velocity in a certain direction).
3. Direction with respect to the constant wind: the mathematical representation of the spectrum and the values of variance and injection length in the longitudinal (parallel to the constant wind) and lateral (perpendicular to the constant wind) directions are different.
4. Magnitude of the nominal airspeed ${v_a}^{nom}$ of the aircraft: necessary to translate the turbulent spectra given in terms of wavenumber κ [rad/m] (in the space domain) into pulsation $\Omega = \kappa {v_a}^{nom}$ [rad/s] (time domain), so to represent the evolution in time of the gust felt by the aircraft.

In the case of one dimensional Von Karman gust field in the vertical plane, flying parallel to the constant wind direction, the spectra of the turbulent velocity are $S_x(\Omega)$ 1 (horizontal) and $S_z(\Omega)$ 2 (vertical).

$$S_x(\Omega) = \frac{{\sigma_x}^2 L_x}{\pi {v_a}^{nom}} \frac{1}{(1 + (1.339 L_x \Omega / {v_a}^{nom})^2)^{5/6}} \qquad (1)$$

$$S_z(\Omega) = \frac{{\sigma_z}^2 L_z}{\pi {v_a}^{nom}} \frac{3 + 8(1.339 L_z \Omega / {v_a}^{nom})^2}{(1 + (1.339 L_z \Omega / {v_a}^{nom})^2)^{11/6}} \qquad (2)$$

The difference between Von Karman and Dryden spectra is in the capability to approximate small scales of turbulence, in favor to the Von Karman one. Since we are interested in energetic scales (the largest of turbulence) this distinction is negligible for the design of the control strategy.

3.2 Flight environment

We are interested in flying at altitudes around 100 meters from the ground, in "countryside" and "small city" environments, where there are not major obstacles such as hills or large buildings. Even in absence of local obstacles, there are many large scale meteorological phenomena, such as air stability [19], that have strong influence and could lead to actual turbulence very different from the modelled one: the control strategy has to be very robust to unexpected gusts. Exaustive tables containing measured values of turbulent variance and injection length are reported in [20]. Notice that they declare an uncertainty of about 30% on the spectrum shape, in particular at low frequencies. Named w_x the turbulent wind component in the direction parallel to the constant wind field (horizontal) and w_z the vertical one, in 'Table 1' the typical parameters for the aforementioned terrain conditions are reported and in 'Figure 1' an example of horizontal spectrum and corresponding gust is shown.

	Standard deviation [m/s]	Injection length [m]
w_x	1.5/2.2	320/425
w_z	0.8/1.2	30/40

Table 1: Standard deviation and injection length in countryside and small city environments, from [20]

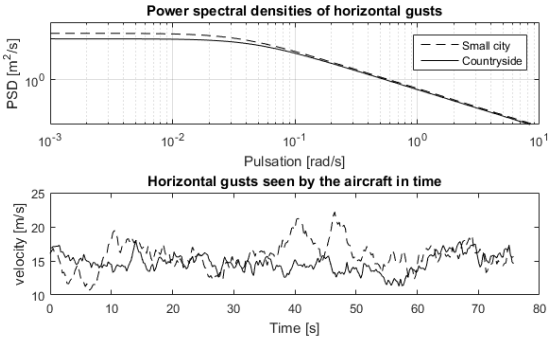


Figure 1: Horizontal wind w_x in frequency and time for 'small city and 'countryside environments

3.3 Scale comparison of turbulence and aircraft

The wingspan of a typical small fixed-wing drone is in the order of 1 meter (for instance in the model used in [12] it is around 4 meters) which is significantly smaller than the injection length scale, thus it is meaningful to model the effect

of energetic turbulence on the plane by its value in a single point [16] (even though smaller turbulent wavelength whose impact changes along the wingspan have to be considered for the stabilization of the aircraft [21]). Moreover we compare the time that is needed by the plane to go through an energetic turbulent structure with the time constant of the slowest longitudinal mode of the aircraft (the phugoid) so to assess if the plane is able to respond to the turbulent input. The energetic turbulence time is obtained by dividing the injection length by the airspeed of the plane (a reasonable value for small UAVs is 20 m/s) thus obtaining 16 s / 21 s for the horizontal turbulent component and 1.5 s / 2 s for the vertical one. The time constant of the plane is calculated as the inverse of the phugoid pulsation for the Langelaan [12] model and it results to be 1.8 s. We conclude that the plane is capable to follow easily horizontal gusts, but it could require a strong control action in order to change its attitude to respond to a vertical gust.

4 Aircraft Model

4.1 Longitudinal dynamics

The longitudinal dynamics equations 13-15 in [12] are used, where the airspeed is separated from the velocity of the wind with respect to the ground. The coefficients are adapted from [12]: for the sake of simplification, we consider a parabolic polar, the thrust aligned with the airspeed and the lift coefficient independent from the derivative of angle of attack. From the longitudinal dynamics in presence of wind, the expression of the total energy 3 (of the longitudinal dynamic) of the aircraft can be obtained, where v_g and h stand for the velocity and the altitude in the ground frame. The energy stored in the rotations is not considered because these should be limited.

$$\begin{aligned} E &= -gh + 1/2 m {v_g}^2 \\ &= -gh + 1/2[(v_a\cos(\gamma) + w_x)^2 + (v_a\sin(\gamma) - w_z)^2] \end{aligned} \tag{3}$$

4.2 Linerarization of the dynamics

The equations are linearized around the stability point given by a nominal airspeed of ${v_a}^{nom} = 20$ m/s, null flight path angle and pitch rate. We use the state vector 4 that is composed (in this order) by the variations from their nominal value of airspeed, flight path angle, angle of attack and pitch rate (four variables that are a common choice to represent the longitudinal dynamics), altitude (which is necessary to compute the potential energy) and integration of the wind gradient in horizontal and vertical directions (because the total energy variation of the aircraft depends on them). Notice that w_x and w_z do not take into account the constant wind field. The input vector 5 is composed by four variables: thrust and elevator commands and the wind gradients in horizontal and vertical directions. While we are able to control the first two, the latter have to be considered as disturbances (with known spectrum). The output vector 6 is the energy, linearized from 3.

$$x = \begin{bmatrix} \delta v_a & \delta\gamma & \delta\alpha & \delta q & \delta h & \delta w_x & \delta w_z \end{bmatrix}^T \tag{4}$$

$$u = \begin{bmatrix} \delta T & \delta e & \dot{w_x} & \dot{w_z} \end{bmatrix}^T \tag{5}$$

$$E_{linear} = -g\delta h + {v_a}^{nom}\delta v_a + {v_a}^{nom}\delta w_x \tag{6}$$

The numerical value of the coefficients is computed from [12] (the full state space matrices are in 'appendix A') and it leads to a poorly damped phugoid, thus, before considering the energy extraction process, a stabilizing controller is needed. This is designed via pole placement technique, the airspeed and the flight path angle (the phugoid mode is a combination of these two variables) are feedbacked through two gains to the elevator in order to obtain a damping of 0.7 (the frequency is kept to its natural value). Via simulations we assess that this stabilizing controller does not interfere with the energy extraction process (the energy of the aircraft oscillates around its nominal value in presence of gust disturbance).

4.3 Phugoid approximation

We investigate if the phugoid approximation of the longitudinal dynamics is valid to represent the effect of the wind on the aircraft. The Bode plots of the transfer functions from the wind gradients to the energy (and to the other state variables) are coincident for all frequencies a part of a small range close to the short period mode pulsation. It is thus reasonable to use the phugoid approximation for the design of the control law and to reduce the order of the system; this result is in accordance with the observations made in [22] . The new state 7 is composed by the airspeed, flight path angle, altitude and the integration of the horizontal and vertical turbulent wind (the state matrices are in 'appendix B').

$$x^{phugoid} = \begin{bmatrix} \delta v_a & \delta\gamma & \delta h & \delta w_x & \delta w_z \end{bmatrix}^T \tag{7}$$

4.4 Energy analysis

From the linear model we obtain the transfer functions from the horizontal and vertical wind to the energy, in 'Figure 2'. The vertical wind to energy transfer functions has the shape of an integrator, where the main contribution to the energy is due to the vertical displacement, while the characteristics of the plane have a small influence on this process. The horizontal wind to energy transfer functions has a low pass behavior whose cut off frequency is close to the phugoid one: we deduce that by tuning the phugoid mode characteristics, the energy extraction process can be modified. In the following, we focus only on the energy extraction process from horizontal gusts and we aim at designing a controller acting on the phugoid mode.

It is possible to compare the effect of the thrust and the horizontal wind gradient. The columns in the input matrix $B^{phugoid}$ (in 'appendix B') corresponding to the thrust (first) and the horizontal wind gradient (third) have the same effect (a part of a minus sign) on the dynamics of the airspeed and

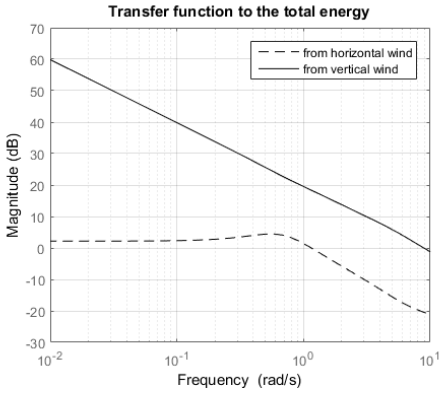


Figure 2: Bode magnitude plots of the transfer functions from the horizontal and vertical wind to the energy

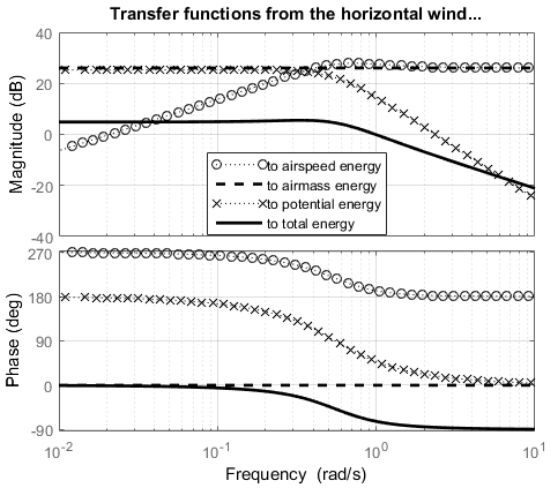


Figure 3: Bode plot of the transfer functions from the horizontal wind gradient to the components of the total energy

the flight path angle, thus allowing us to treat wind gradients as a source of additional propulsion. Their effect on the overall energy of the plane is neverthless different, since the integration of wind, but not the one of the thrust appears in 3. The same conclusion is obtained experimentally in [23].
The linear expression of the energy 6 is composed by three terms that we name, slightly modifying the convention in [4]:

1. Airspeed energy: $E_{airspeed} = {v_a}^{nom}\delta v_a$
2. Potential energy: $E_{potential} = -g\delta h$
3. Airmass energy: $E_{airmass} = {v_a}^{nom}\delta w_x$

When a stabilizing controller is used to keep a constant airspeed, all the energy coming from the wind will be converted into altitude gain. From 'Figure 3', we assess which is the dominant contribution at a certain pulsation of the gust and how energy passes from one component to the other.

1. Airmass energy is a constant gain so its contribution is constant over all the frequencies.
2. Airspeed energy shows a high pass behavior (whose cut-off frequency is the phugoid one) so to counterbalance the airmass energy at high frequencies (the wind variation is seen only as a variation of airspeed, since the inertial velocity of the plane does not have time to change).
3. Potential energy has a low pass behavior and it counterbalances the airmass energy at low frequencies (the controller is able to stabilize the airspeed around its nominal value).

4.5 Static gain of the wind-to-energy transfer function

The pole placement controller (used before for stabilization) allows us to change the phugoid pulsation, while keeping its damping at 0.7. Numerically we find a relation between the static gain of the horizontal wind-to-energy transfer function and the phugoid frequency, as in 'Figure 4', and its effect in the time domain is shown in 'Figure 5'. This property allows us to act on the shape of the horizontal wind-to-energy transfer function by tuning a feedback controller from airspeed and flight path angle to the elevator ('Figure 6'), with an approach similar to the one used in H-infinity technique.

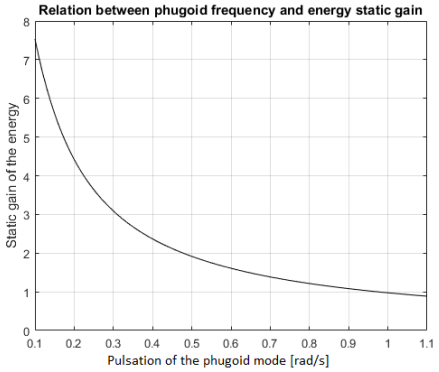


Figure 4: Relation between the static gain of the wind-energy transfer and the phugoid frequency (given a damping of 0.7).

5 Control Strategy

5.1 Overview on the control architecture

The objective of the control law is extracting energy from gusts. The innovation present in this work consists in using a feedback law that switches between two sets of gains depending on the energy derivative of the aircraft to control the elevator action, as in 'Figure 7'.
When the aircraft is gaining energy, we exploit it as much as possible, while, when it loses it, we aim at losing the smallest quantity. This translates into having large oscillations of energy (high gain of the wind-energy transfer function) when it

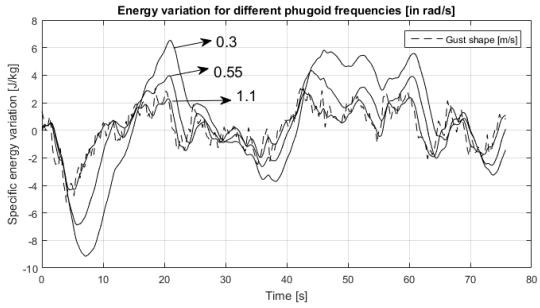


Figure 5: Effect of tuning the phugoid frequency in the time domain.

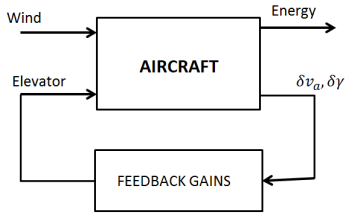


Figure 6: Representation of the system and the pole placement controller.

is entering the plane and very small ones when it is exiting the system (small gain of the wind-energy transfer function). Notice that we do not try to invert the natural energy exchange process (from losing to gaining energy) but we act only on the amplitude of it.

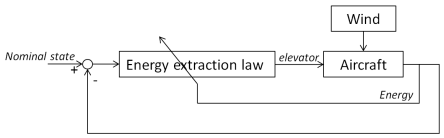


Figure 7: Overview of the architecture.

5.2 *Phugoid pole placement technique*

We exploit the relation found between the phugoid frequency and the static gain of the system in 'Figure 4' to achieve the aforementioned objective. Since the turbulent injection frequency is sensibly lower than the frequency range where we move the phugoid poles, the static gain corresponds to the gain at the injection frequency. To place the phugoid frequency the pole placement method of a second order system is used, and the gains become dependent on the energy derivative 8, where ${k_1}^{in/out}$ represents the gain computed respectively for the "entering" (in) and "exiting" (out) energy law and E is 3. The same rule is valid for the gain k_2. The magnitude Bode plots of the two switched subsystems are in 'Figure 8' and the schematic representation of this law is in 'Figure 9'.

$$\delta e = k_1(dE/dt)\ \delta v_a + k_2(dE/dt)\ \delta\gamma$$
$$k_1(dE/dt) = \begin{cases} {k_1}^{in} & \text{if } dE/dt \geq 0 \\ {k_1}^{out} & \text{if } dE/dt < 0 \end{cases} \quad (8)$$

The set of phugoid frequency pairs $(\omega_{in}, \omega_{out})$ among whom we can choose for our control strategy is limited by the validity of the phugoid approximation (in the direction of higher frequencies), by the non-linear effects present in the realistic model (for lower frequencies) and by the stability of the switched system (when the difference between the two frequencies becomes too large, as it will be discussed in 'subsection 5.4').

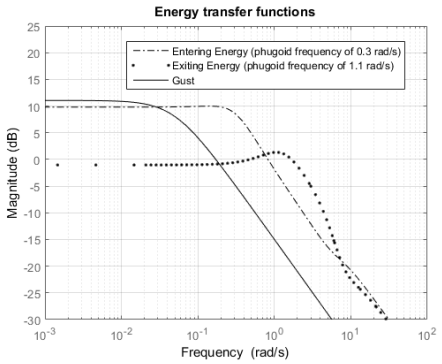


Figure 8: Magnitude of the closed loop transfer functions between the horizontal turbulent wind and the energy (dotted lines) when we place the phugoid poles, for instance, at 0.3 rad/s ("entering" energy mode) and 1.1 rad/s ("exiting" energy mode). The solid line is the transfer function representing the Dryden turbulent spectrum.

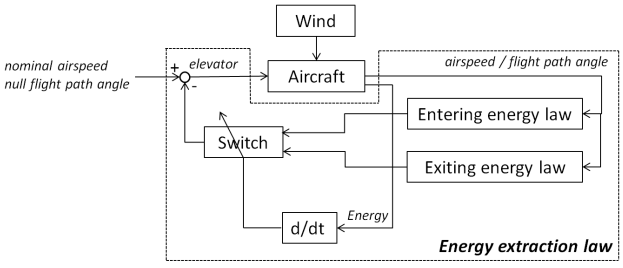


Figure 9: Energy extraction control strategy.

5.3 *Switched system formalism*

We can classify the controlled system that is obtained as belonging to the class of autonomous switched system with externally forced switching [24]. Turbulence is an external disturbance, the elevator action is hidden in the modification of the phugoid dynamics and the thrust input is not used, thus the only way to control this system is to act on the switching signal. Moreover, the mode sequence is fixed, since we are switching between only two systems (even though it would

have been possible to design more than two modes, we chose two to stick with physical reasoning). The switched system can be represented by 9, where the states are continuous across switching instants and $\sigma(t)$ is the switching signal. In our case $\sigma(t)$ is not an explicit function of time but of the states and it can only assume binary values (in or out), meaning the system is in the "entering" or "exiting" energy mode. We introduce the notation ${A_{\sigma(t)}}^{phugoid}$ 10 to represent a matrix that assumes the values of the phugoid state matrix with pulsation ω_{in} or ω_{out} depending on $\sigma(t)$.

$$\dot{x} = \begin{bmatrix} {A_{\sigma(t)}}^{phugoid} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & -20 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \end{bmatrix} x + \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{w}_x \tag{9}$$

$$A_{\sigma(t)}^{phugoid} = \begin{cases} A_{in}^{phugoid} & \text{if } dE/dt \geq 0 \\ A_{out}^{phugoid} & \text{if } dE/dt < 0 \end{cases} \tag{10}$$

5.4 *Switched system stability*

We are interested in guaranteeing the stability of the airspeed and the flight path angle dynamics (since our objective is to have an "unstable" altitude) in the case of arbitrary switching, in fact, even if the two subsystems between whom we switch are stable, this is not sufficient to prove the stability of the overall switched system. We apply theorem 3.1 in [25], which gives a necessary and sufficient condition to have a Common Lyapunov Quadratic Function for a pair of switched second order stable linear time invariant systems (such as the phugoid representation). In 'Figure 10' the stability domain is shown, depending on the phugoid pulsations for the entering and exiting energy case. We choose the stable couple $(\omega_{in}, \omega_{out}) = (0.3, 1.1)$ rad/s in order to perform simulations, but we cannot assess apriori its performances in terms of energy extraction (there could be stable couples giving better performances).

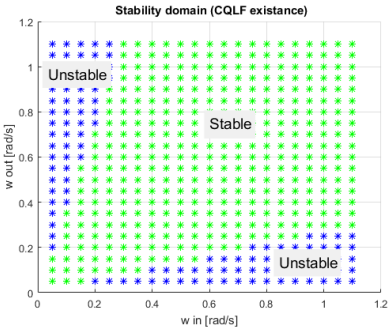


Figure 10: Stability domain of the switching system, expressed in function of the entering (in) and exiting (out) phugoid pulsations.

6 Simulations

6.1 *Energy evolution in time*

The capability of our strategy to extract energy from gusts is assessed by simulations on a non-realistic sinusoidal gust of low frequency (0.05 rad/s, in 'Figure 11') and on realistic Von Karman gusts. In both cases we are able to obtain an increase in energy with respect to the stabilizing controller. A great advantage of our technique is that the control action is not significantly increased (and so the energy of the battery cunsumed by the actuators) with respect to the one needed for stabilisation purposes only ('Figure 12'), contrary to the results in [12].

We can simulate on the same gust different pairs of phugoid frequencies ('Figure 13'); as expected from 'Figure 4', when the difference between the entering and exiting phugoid frequencies is increased, the extracted energy increases as well. Further insights are necessary to understand the relation between the chosen frequencies and the achievable energy increase.

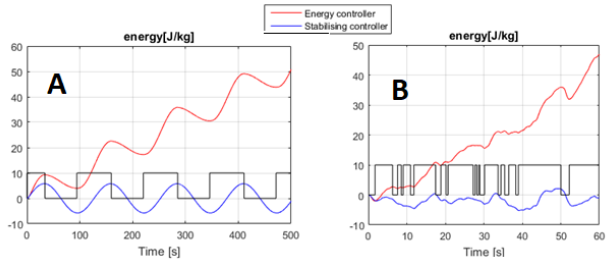


Figure 11: Energy evolution and switching signal in the case of sinusoidal (A) and Von Karman (B) gust. The "energy extraction" controller is compared to a simple stabilizing controller. The black line is the switching signal (a relay could be needed to avoid fast switching).

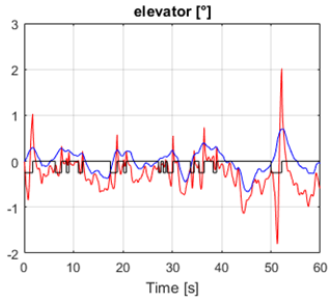


Figure 12: Elevator deflection for the energy extraction controller (red) compared to its usage by the stabilizing controller (blue).

6.2 *Statistical simulations*

We simulate our control strategy applied to the non-linear longitudinal rigid aircraft model on a set of Von Karman horizontal gusts, for the phugoid frequencies chosen in 'subsection 5.4'. A set of 200 gusts (100 representative of "countryside" environment and 100 of the "small city" one) of 60 seconds each are used ('Table 2'). We can estimate the achievable reduction of thrust ΔT from the amount necessary to fly in still air (where the airspeed is the ground speed too) corre-

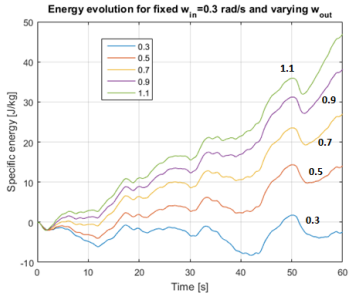


Figure 13: Effect of the variation of ω_{out} for the same gust.

sponding to the extracted power $P_{extracted}$ in 11.

$$P_{extracted} = \Delta T {v_a}^{nom} \tag{11}$$

	Countryside	Small city
Extracted energy [J/Kg]	40.25 ± 7.98	51.73 ± 8.23
Extracted power [N/Kg]	0.67 ± 0.133	0.86 ± 0.137
Thrust reduction [%]	6.22 ± 0.013	7.70 ± 0.013

Table 2: Extracted power and achievable thrust reduction

6.3 *Phase plane*

Since the phugoid mode is two-dimensional, we use a phase plane technique to analyze it ('Figure 14'). A switching surface (close to the null variation of airspeed axis) can be identified, seemingly regardless of the evolution of the turbulence disturbance. Even though further insights are needed, this result is linked to the similarity of the transfer functions from the wind to the energy derivative and the airspeed variation: they show identical phase plots but shifted by 180, thus the sign of the airspeed is always the opposite of the one of the energy derivative. Thanks to this nice property, the switch signal can be based on the airspeed evolution, thus reducing the number of sensors needed (there is no need to measure the total energy).

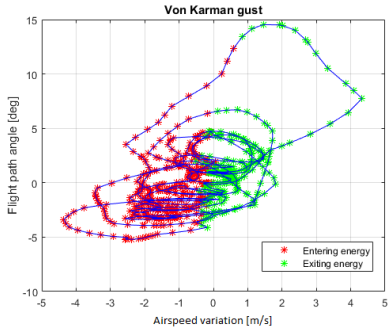


Figure 14: Phase plane of airspeed and flight path angle for Von Karman gust.

6.4 *Energy gain dependence on the frequency of the gust*

The control strategy that we developed is based on the difference on static gains between the entering and exiting energy transfer functions, that is at frequencies consistently lower than the phugoid one. In the frequency interval where the phase of the transfer functions is varying, we cannot foresee the performances of the control law. When the frequency of the gust gets consistently higher than the phugoid the "entering" and "exiting" transfer functions in 'Figure 8' are coincident, thus we expect that no energy extraction is achievable for that range. To check these observations, the whole controlled system is treated as input-output between the gust and the energy, and we scan on a set of gust frequencies the power extracted by the aircraft ('Figure 15'). A cutoff frequency is clearly identified and it is close to the phugoid frequency of the aircraft, meaning that we can extract energy from low frequency gusts and not from high frequency ones.

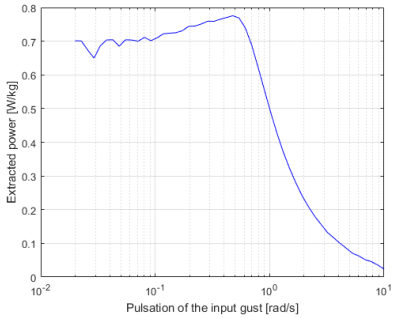


Figure 15: Power extraction capability depending on the frequency of the gust input.

7 Conclusions and Future Work

First, we clearly presented the atmospheric gusty environment and described how much the gust spectrum can fit to the reality. Then we investigated the energy exchange mechanism between the wind and the aircraft. We proposed an innovative energy extraction control strategy based on a switching controller and we analyzed it. Energy savings between $5-10\%$ of the thrust needed during straight flight are achievable for realistic horizontal gusts and proven stability of the aircraft. Room for improvement is available, since it seems possible from numerical simulations to relax the stability limitations. This result is obtained without increasing considerably the control activity with respect to a stabilizing controller and with only the need of two sensors, one measuring the airspeed and one the flight path angle.

References

[1] W.H.Phillips. Propulsive effects due to flight through turbulence. *Journal of Aircraft*, 12(7):867–871, 1975.

[2] T.Kiceniuk. Dynamic soaring and sailplane energetics. *Technical Soaring*, 25(4):221–227, 2001.

[3] T.Kiceniuk. Calculations on soaring in sink. *Technical Soaring*, 25(4):228–230, 2001.

[4] J.Bailey. Total energy control concepts applied to flight in windshear. In *AIAA Guidance, Navigation and Control conference, AIAA 1987-2344*, 1987.

[5] C.M.Belcastro and A.J.Ostroff. Total energy-rate feedback for automatic glide-slope tracking during wind-shear penetration. In *NASA technical paper 2412*, 1984.

[6] A.Lambregts. Vertical flight path and speed control autopilot design using total energy principles. In *AIAA Guidance and Control conference, AIAA 83-2239*, 1983.

[7] Z.Akos M.Nagy S.Leve and T.Vicsek. Thermal soaring flight of birds and unmanned aerial vehicles. *Bioinspiration and Biomimetics*, 5(4):045003, 2010.

[8] G.Sachs. Minimum shear wind strength required for dynamic soaring of albatrosses. *International Journal of Avaian Science*, 147(1):1–10, 2004.

[9] N.De Divitiis. Effect of microlift force on the performance of ultralight aircraft. *Journal of Aircraft*, 39(2):318–332, 2002.

[10] H.U.Mai. The effect of aeroelasticity upon energy retrieval of a sailplane penetrating a gust. *Technical soaring*, 10(4):62–72, 1985.

[11] N.Gavrilovic E.Benard P.Pastor and J.M.Moschetta. Performance improvement of small uavs through energy-harvesting within atmospheric gusts. In *AIAA Atmospheric Flight Mechanics Conference, AIAA 2017-1630, Grapevine, TX, USA*, 2017.

[12] J.W.Langelaan. Biologically inspired techniques for small and micro unmanned air vehicles. In *AIAA Guidance, Navigation and Control Conference and Co-located Conferences AIAA 2008-6511*, 2008.

[13] N.T.Depenbusch and J.W.Langelaan. Receding horizon control for atmospheric energy harvesting by small uavs. In *AIAA Guidance, Navigation and Control Conference, AIAA 2010-8177*, 2010.

[14] P.B.S.Lissaman and C.K.Patel. Neutral energy cycles for a vehicle in sinusoidal and turbulent vertical gusts. In *45th AIAA Aerospace Science Meeting and Exhibit, AIAA 2007-86*, 2007.

[15] H.Lee C.K.Patel and I.M.Kroo. Extracting energy from atmospheric turbulence. In *XXIX OSTIV Congress*, 2008.

[16] B.Etkin. *Dynamics of atmospheric flight, Chapter 13: Flight in a turbulent atmosphere*. John Wiley and sons, Toronto, 1972.

[17] I.Van Der Hoven. Power spectrum of horizontal wind speed in the frequency range from 0.0007 to 900 cycles per hour. *Journal of meteorology*, 14:160–164, 1956.

[18] T.Von Karman. Progress in the statistical theory of turbulence. *Proceedings of the National Academy of Sciences*, 34:530–540, 1948.

[19] J.C.Kaimal J.C.Wyngaard Y.Izumi and O.R.Coté. Spectral characteristics of surface-layer turbulence. *Quart. J.R. Met. Soc.*, 98(417):563–589, 1972.

[20] ESDU Data Item 85020. *Characteristics of atmospheric turbulence near the ground. Part II: single point data for strong winds (neutral atmosphere)*. Engineering Science Data Unit, London, UK, 2001.

[21] S.Watkins M.Thompson B.Loxton and M.Abdulrhaim. On low altitude flight through the atmospheric boundary layer. *International Journal of Micro Air Vehicles*, 2(2):55–68, 2010.

[22] P.P.Sukumar and M.S.Selig. Dynamic soaring of sailplanes over open fields. In *28th AIAA Applied Aerodynamics Conference, AIAA 2010-4953*, 2017.

[23] M.Abdulrahim and S.Watkins. Dynamic sensitivity to atmospheric turbulence of a fixed-wing mav with varying configuration. In *AIAA Guidance, Navigation and Control conference, AIAA 2009-5907*, 2009.

[24] F.Zhu and P.J.Antsaklis. Optimal control of switched hybrid systems: a brief survey. In *Technical report of the ISIS Group at the Univeristy of Notre Dame, ISIS-2013-007*, 2013.

[25] R.N.Shorten and K.S.Narendra. Necessary and sufficient conditions for the existence of a common quadratic lyapunov function for a finite number of stable second order linear time-invariant systems. *International Journal of adaptive control and signal processings*, 16(10):709–728, 2002.

APPENDIX A: COMPLETE LINEAR MODEL

The following matrices are the numerical values of the state space representation when considering the full linear model.

$$A^{complete} = \begin{bmatrix} -0.049 & -9.81 & -1.79 & -0.061 & 0 & 0 & 0 \\ 0.049 & 0 & 6.78 & 0.023 & 0 & 0 & 0 \\ -0.049 & 0 & -6.78 & 0.98 & 0 & 0 & 0 \\ 0 & 0 & -31 & -2.57 & 0 & 0 & 0 \\ 0 & -20 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B^{complete} = \begin{bmatrix} 1 & -0.119 & -1 & 0 \\ 0 & 0.453 & 0 & 0.05 \\ 0 & -0.453 & 0 & -0.05 \\ 0 & -49.2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C^{complete} = \begin{bmatrix} 20 & 0 & 0 & 0 & -9.81 & 20 & 0 \end{bmatrix}$$

APPENDIX B: PHUGOID APPROXIMATION OF THE DAMPED AIRCRAFT

The phugoid approximation of the damped system is obtained by reducing the (fast) short period mode states, which is a standard technique. We obtain the following state space matrices.

$${A_{damped}}^{phugoid} = \begin{bmatrix} -0.044 & -9.6 & 0 & 0 & 0 \\ 0.029 & -0.73 & 0 & 0 & 0 \\ 0 & -20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B^{phugoid} = \begin{bmatrix} 1 & 1.18 & -1 & 0 \\ 0 & -6.7 & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C^{phugoid} = \begin{bmatrix} 20 & 0 & -9.81 & 20 & 0 \end{bmatrix}$$

Prioritized Control Allocation for Quadrotors Subject to Saturation

E.J.J. Smeur*, D.C. Höppener, C. De Wagter
Delft University of Technology, Kluyverweg 1, Netherlands

ABSTRACT

This paper deals with the problem of actuator saturation for INDI (Incremental Nonlinear Dynamic Inversion) controlled flying vehicles. The primary problem that arises from actuator saturation for quadrotors, is that of arbitrary control objective realization. We have integrated the weighted least squares control allocation algorithm into INDI, which allows for prioritization between roll, pitch, yaw and thrust. We propose that for a quadrotor, the highest priority should go to pitch and roll, then thrust, and then yaw. Through an experiment, we show that through this method, and the appropriate prioritization, errors in roll and pitch are greatly reduced when applying large yaw moments. Ultimately, this leads to increased stability and robustness.

1 INTRODUCTION

Control allocation is often described as the problem of distributing control effort over more actuators than the number of controlled variables [1, 2, 3]. This is something that occurs in traditional aircraft as well as drones, such as hexarotors and octorotors. What sometimes does not receive a lot of attention, is that the problem of how to deal with actuator saturation is also part of the control allocation topic and, in some cases, can be very important.

Especially for aerial vehicles with coupled control effectors, such as quadrotors, actuator saturation may lead to undesired, or if occuring for longer timespans, even catastrophic behaviour. It may be that the desired thrust, and/or control moments in roll, pitch and yaw, can not be achieved due to actuator saturation. In absence of an adequate control allocation algorithm, it is left to chance which part of the control objective will suffer, it may be the thrust, roll, pitch, or yaw.

However, for the flight stability of multirotor vehicles, it is far more important to apply the right roll and pitch control moments than to apply the right yaw moment, since the thrust vector is indifferent to the yaw in body axis. Therefore, we would like the control allocation algorithm to *prioritize* the control objective of roll and pitch over that of yaw, and to calculate the control inputs accordingly.

In previous research, we have developed an Incremental Nonlinear Dynamic Inversion (INDI) controller for Micro Air Vehicles (MAV) [4, 5]. We have shown that this control method is very good at disturbance rejection and needs little model information. Moreover, we presented a method to include the effects of propeller inertia, yielding faster and more accurate yaw control. This aggressive yaw control can easily lead to saturation of multiple actuators, especially when commanding large yaw changes. These saturations often lead to errors in roll and pitch angles and in the thrust, causing the vehicle to lose control of its position and potentially crash.

But also external moments, such as wind disturbances, or actuator faults can lead to saturation. This is why a control allocation method needs to be added to the INDI control structure. Multiple control allocation algorithms have been proposed, some of which do not adequately address prioritization: ganging, redistributed pseudo-inverse, direct control allocation; and some of which do: linear programming and quadratic programming [6]. In this paper, we will consider a quadratic cost function, and the corresponding quadratic optimization problem. A solution to this problem can be found in a straightforward way using the active set method, as has been shown by Härkegård [7].

In this paper, we integrate the Weighted Least Squares (WLS) control allocation algorithm into the INDI attitude controller. Further, we show through an experiment that prioritization of roll and pitch over yaw leads to stability improvements. The structure of this paper is as follows: first, the INDI control law is introduced in Section 2. Second, Section 3 elaborates on the WLS method and how it integrates with INDI attitude control. Third, the experimental results are presented in Section 4, and we end with conclusions and future work in Section 5.

1.1 Related Work

As opposed to our approach of prioritization, some research has focused on the preservation of control *direction* [8, 3]. This means that in case of saturation, a solution for the actuator inputs is sought that corresponds to a linear scaling of the original control objective. This approach may be useful for systems where all axes are equally important. However, for a quadrotor, if a large yawing moment is needed, the actuators can easily saturate due to the low control effectiveness in this axis. Scaling the desired control moments will make the roll and pitch control suffer, which may lead to instability.

Recently, Faessler et al. implemented a heuristics based algorithm for priority management [9]. They showed that prioritizing roll and pitch over yaw can lead to stability improve-

*Email address: e.j.j.smeur@tudelft.nl

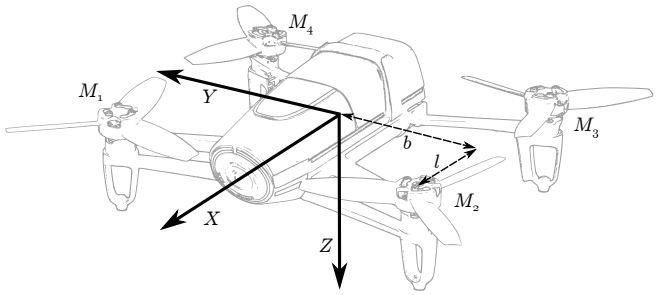


Figure 1: Axis definitions.

ments. However, the suggested algorithm resembles the Redistributed Pseudo Inverse method (RPI), which is known in some cases to not find the control solution even if the control objective is achievable [10]. Furthermore, the scheme is particularly constructed for quadrotors, and does not generalize.

The WLS approach is much more general, as it does not depend on a certain configuration of actuators. The method has been suggested for quadrotors by Monteiro et al. [11], but was only implemented in simulation. Furthermore, the weighting matrix, that determines the priorities in the cost function, is not discussed.

2 Incremental Nonlinear Dynamic Inversion and Actuator Saturation

In previous work [4], we derived INDI control for MAVs. A detailed derivation is beyond the scope of this paper, but the main feature of the controller is its incremental way of controlling angular accelerations. The basic idea is that the current angular acceleration is caused by the combination of inputs and external moments. In order to change the angular acceleration, all that is needed is to take the previous inputs and increment them, based on the error in angular acceleration and the control effectiveness.

A distinction is made between two types of forces and moments: those that are produced by inputs, and those that are produced by changes in inputs. The forces and moments produced due to the propeller aerodynamics fall in the first category, and the torque it takes to spin up a propeller falls in the second category. Both need to be accounted for in different ways, which is why the control effectiveness matrix is split up in two parts: $G = G_1 + G_2$, where G_2 accounts for the propeller spin up torque. Though the algorithms presented here have broad applicability, we will, in order to promote clarity, consider the quadrotor shown in Figure 1, with the illustrated axis definitions. We define the angular rotation vector Ω, its derivative $\dot{\Omega}$ and the angular rate of the propellers ω. Then, if we assume a linear control effectiveness and that gyroscopic effects of the vehicle can be neglected [4], the system equation in incremental form is

$$\dot{\Omega} - \dot{\Omega}_0 + G_2 L(\omega - \omega_0) = (G_1 + G_2)(\omega - \omega_0), \quad (1)$$

subject to

$$\omega_{\min} \leq \omega \leq \omega_{\max}, \quad (2)$$

where L is the lag operator, e.g. $\omega(k-1) = L\omega(k)$. Note that the angular acceleration needs to be obtained by deriving it from gyroscope measurements through finite difference. This signal can be quite noisy, and will need appropriate filtering. In order to synchronize all signals with subscript 0, they all need to be filtered with this same filter.

Equation 1 can be turned into a control law using the matrix inverse or the pseudo-inverse:

$$\omega = \omega_0 + (G_1 + G_2)^{-1}(\nu - \dot{\Omega}_0 + G_2 L(\omega_c - \omega_0)), \quad (3)$$

but calculating the control input like this does not guarantee satisfying Equation 2. If the control inputs exceed the bounds, simply clipping them will result in different control moments than desired.

Instead, Equation 3 is replaced with a method that calculates the control inputs while respecting the limits and prioritization. This can be done with a weighted least squares (WLS) optimization. Since our system description (Equations 1 and 3) is in incremental form, we will first write it as a standard least squares problem through a change of variables:

$$v = Gu \quad (4)$$

subject to

$$u_{\min} \leq u \leq u_{\max}. \quad (5)$$

where the control objective is $v = \dot{\Omega} - \dot{\Omega}_0 + G_2 L(\omega - \omega_0)$, and the input is $u = \omega - \omega_0$. The limits $u_{\min}$ and $u_{\max}$ follow from these definitions and Equation 2.

3 Using the Active Set Method to Solve the Constrained Allocation Problem

Though in this paper we will apply the algorithm to a quadrotor, for the control allocation we will also consider over-actuated systems. This means that we have to include a cost for actuator usage in the cost function, such that there is only one optimum. This will make the derived methodology easily applicable to other systems, like multirotors with more than four rotors, or some over-actuated hybrid systems like the Quadshot [12].

In most cases, we would like to formulate the control allocation problem as a sequential least squares problem. Primarily, we want to minimize the error between the control objective and the angular acceleration increment produced by the calculated control increment. This can be captured in a first cost function. Secondly, given the inputs that minimize the primary cost function, we would like the actuators to spend the lowest amount of energy possible. If G has full rank, the secondary cost function can be omitted, as the primary cost function will only have one solution. However, when there are more actuators than control objectives, the second cost function will minimize expended energy and avoid actuators steering in opposite directions.

The sequential least squares problem is more difficult to solve than a least squares problem with a single cost function. This is why we adopt the WLS problem formulation from Härkegård [7], where the cost for errors in the control objective is combined with a cost for applying inputs:

$$\begin{aligned} C(u) = &\quad \|W_u\ (u - u_d)\|^2 + \gamma \|W_v\,(Gu - v)\|^2 \\ = &\quad \left\| \begin{pmatrix} \gamma^{\frac{1}{2}} W_v\, G \\ W_u \end{pmatrix} u - \begin{pmatrix} \gamma^{\frac{1}{2}}\, W_v\, v \\ W_u\, u_d \end{pmatrix} \right\|^2, \end{aligned} \tag{6}$$

where W_v is the diagonal weighting matrix for the control objective, and W_u is the diagonal weighting matrix for the inputs. The distinction between the primary and secondary objective is made by the scale factor $\gamma >> 1$. For convenience, we define

$$A = \begin{bmatrix} \gamma^{\frac{1}{2}} W_v\,(G_1 + G_2) \\ W_u \end{bmatrix} \text{ and } b = \begin{bmatrix} \gamma^{\frac{1}{2}} W_v v \\ W_u u_d \end{bmatrix}. \tag{7}$$

Now that the problem is formulated as a regular quadratic programming problem, it can be solved using the well known active set method [7, 13, 14], to find the inputs that minimize the cost function. The algorithm divides the inputs into a free set and an active set, which correspond to the inputs that are not saturated and to the actuators that are saturated respectively. The method disregards the inequality constraints for the free set, and for the active set W treats the constraints as equality constraints. At every iteration, it is evaluated if the division between active and free set is correct. For completeness, we explain our implementation of the active set method in Algorithm 1.

The algorithm stops when the solution is optimal, or a maximum number of iterations is reached. Though the algorithm is guaranteed to find the optimum in a finite number of iterations, we may impose a maximum number of iterations that can be executed in a real time application. If the algorithm stops because the maximum number of iterations is reached, the solution will not be optimum. However, since the value of the cost function decreases at each iteration [14], the result will be better than at the start of the algorithm.

3.1 Particularities for WLS applied to INDI

Since we are applying the WLS control allocation scheme to the INDI controller, the inputs are incremental. This means that the bounds on the input (increment) change every time step, and the solution for the increment at one time may not be feasible the next time step. The initial guess for the input, u^0, can therefore not be the solution of the previous time, as is often done for non-incremental controllers [7, 13, 6]. Instead, we take as initial input the mean of the maximum and minimum input increment:

$$u^0 = \frac{1}{2}(u_{\max} - u_{\min}). \tag{13}$$

Additionally, if we consider an over-actuated system, the choice of the preferred increment u_p becomes important, as

Algorithm 1: Active set method for WLS problem

Initialization:
$W = \{\emptyset\}, \quad u^0 = (u_{\max} - u_{\min})/2, \quad d = b - Au^0,$
$S = [\emptyset]$
for $i = 0, 1, 2, .., n_{\max}$ **do**
  Determine the free columns in A:

$$A_f = A\,(:, h)\,, \quad h \notin W \tag{8}$$

  Determine the optimal perturbation by solving the following least squares problem for p_f:

$$d = A_f p_f \tag{9}$$

  Now p is constructed from p_f with zeros for the elements that are in W.
  if $u^i + p$ is feasible **then**
    $u^{i+1} = u^i + p$ and: $d = d - A_f p_f$
    The gradient and Lagrange multipliers are computed with:

$$\nabla = A^T d \text{ and: } \lambda = S\nabla \tag{10}$$

    if all $\lambda \geq 0$ **then**
      The solution u^{i+1} is optimal $u = u^{i+1}$;
    else
      The constraint associated with the most negative λ has to be removed from the active set W. Re-iterate with this active set.
  else
    The current solution violates a constraint which is not in W. Determine the maximum factor α such that αp is a feasible perturbation, with $0 \leq \alpha < 1$. Update the residual d and the solution u^{i+1}:

$$u^{i+1} = u^i + \alpha p \tag{11}$$

$$d = d - A_f \alpha p_f \tag{12}$$

    Finally, update the active set and store the sign of the constraint: $S_{jj} = \text{sign}\,(p_j)$ with j the index of the new active constraint.

there is some degree of freedom in choosing the inputs that will produce the required forces and moments. Some of these combinations may require more energy than what is optimal, for instance if two actuators counteract each other in order to produce a net zero output. Clearly, this can be achieved more efficiently by giving zero input to both actuators. For non-incremental controllers, this means that that u_p is a zero vector. For an incremental controller, this means that $u_p = u_{\min}$, assuming that the actuators produce zero force/moment at $u_{\min}$.

3.2 Choice of Weighting Matrices

As for any optimization, the result entirely depends on the choice of the cost function. In this case, we have the freedom to choose W_v, W_u and γ.

For W_v, we choose the diagonal elements to be 1000, 1000, 1 and 100 for roll, pitch, yaw and thrust respectively. The reason that we give roll and pitch a higher priority than thrust, is because the thrust can only be applied in the right direction if the vehicle has the right attitude. As an example, suppose that the quadrotor is inverted. With the thrust vector pointing down, it will lose altitude fast. The controller will have to flip the airframe, and increase thrust to climb. However, if priority would be put on the thrust, the vehicle could, in the extreme case, never change the attitude, as all motors would have to give full thrust.

In general, it appears that satisfying (part of) the roll and pitch objectives, will lead to a reduction of said objectives in the short term, as it typically does not take long to rotate to a desired attitude. On the other hand, satisfying (part of) the thrust objective, might not lead to a reduction of this objective in the short term, as the thrust vector may be pointing in the wrong direction or a large continuous thrust may be needed over a long period of time. Therefore to the authors, prioritizing pitch and roll over thrust seems to be the most stable configuration. However, for a specific quadrotor, the best prioritization scheme may depend on the mission profile.

We choose $\gamma^{\frac{1}{2}} = 10000$ and for W_u we take the identity matrix, since all actuators are 'equal'. Do note that the relative scaling of the signals u and v plays a role here. Also note that, even though we give a lower weighting to some signals, they can still become dominant in the cost function if no bounds are applied. As an example, consider a quadrotor that has to climb five kilometers. In case of a simple PD controller without bounds, an enormous thrust will be commanded, leading to more cost in Equation 6.

3.3 Computational Complexity

The computational complexity of the active set algorithm scales with the number of actuators in two ways. First, each additional actuator will add a row and a column to the matrix A, and therefore increase the computational complexity of solving the quadratic problem each iteration of the active set algorithm. Additionally, if there are more actuators, more actuators can saturate in different combinations. This may lead to more iterations on average, as well as more iterations in a worst case scenario.

An analysis of the performance of the active set algorithm on a benchmark problem set, with control objectives in $\mathbb{R}^3$ was done by Petersen and Bodson [13]. They report that the method is efficient in case of few actuators, but that it does not scale well with the problem size. Specifically, for 15 actuators or more, an interior point method is more efficient. Since our control objective is in $\mathbb{R}^4$, this point can be somewhere else.

Clearly, it is very beneficial for the computational performance to have few actuators. If computational time is a problem, it might be an option to combine several actuators into single 'virtual' actuators, often referred to as 'ganging'.

However, we are able to run the WLS scheme on an STMF4 microprocessor, which is equipped with a floating point unit, for four actuators at 512 Hz without any problem. Our implementation uses single precision floating point variables.

4 EXPERIMENTS

As mentioned in the introduction, actuator saturation often occurs due to yaw commands, as the yaw moment generation of the actuators is relatively weak. Without proper priority management, this is a case where instability can occur. In order to demonstrate the ability of the WLS control allocator to improve stability of the vehicle through priority management, an experiment is performed.

The hypothesis is that the WLS control allocation scheme, with the prioritization as defined in section 3.2, improves the tracking of pitch and roll when large yaw moments are required, as compared to calculating the inputs with the pseudo-inverse and clipping the result.

To test for this hypothesis, the hovering drone will be given an instant step in its heading reference of 50 degrees. This is enough to cause severe actuator saturation. The drone is controlled by a pilot, who will bring the drone back to the hovering position after each maneuver. During the maneuver, the pilot does not give any commands.

4.1 Experimental Setup

The test is performed using a Bebop 1 quadrotor from Parrot, running the Paparazzi open source autopilot software. The Bebop is equipped with an internal RPM controller, which accepts commands between 3000 and 12000 RPM. In practice, we found that in static conditions the motors saturate well before 12000 RPM. To avoid commands above the saturation limit that will not have any effect, the limit in the software is put at 9800 RPM.

Again, for details on the INDI control algorithm employed, we refer to our previous papers [4, 5]. However, we will list the parameters used for the experiment. Prior to the experiment, the following control effectiveness matrices were identified through test flights:

$$G_1 = \begin{bmatrix} 18 & -18 & -18 & 18 \\ 11 & 11 & -11 & -11 \\ -0.7 & 0.7 & -0.7 & 0.7 \\ -0.4 & -0.4 & -0.4 & -0.4 \end{bmatrix} \cdot 10^{-3} \quad (14)$$

$$G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -65 & 65 & -65 & 65 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot 10^{-3} \quad (15)$$

The filter that is used for the angular acceleration is a second order Butterworth filter with a cutoff frequency of 5 Hz.

4.2 Results

Figure 2 shows the results of the experiment for the pseudo-inverse on the top and for WLS on the bottom. From the left, the first three figures on each row show the Euler angles for 15 and 12 repetitions of the experiment for the pseudo-inverse and WLS respectively. For WLS, two repetitions were rejected, because the pilot steered during the yaw step. The last figure on each row shows the inputs to the actuators during the first repetition.

First, from the plot of the ψ angle it can be observed that with WLS there is no overshoot, but the rise time is longer. The longer rise time can be explained, because for WLS, the inputs are not saturated the whole time the vehicle is moving towards the reference. Because of this, for WLS, the angular velocity does not become as high and the quadrotor is able to reduce the angular velocity without saturating the actuators. For the pseudo-inverse, the situation can be compared with integrator windup. The quadcopter builds up so much angular velocity while the actuators are saturated, that when it has to reduce this angular velocity, the actuators saturate in the other direction and the vehicle overshoots.

Though now it may seem that WLS solves this problem, this is not the case. The figure merely shows that due to the prioritization, the vehicle can not accelerate as fast in the yaw axis, which is why the overshoot does not occur. For larger heading changes, when the vehicle will accumulate angular velocity in the yaw axis over a longer time, overshoot is also observed.

However, the plots of pitch and roll show the merit of the WLS control allocation (note the different scale). To condense this information, we consider the maximum deviation of the roll and pitch angle from zero as a measure of the performance for each repetition. The mean and standard deviation of this maximum error per repetition is presented in Table 4.2.

Clearly there is a very significant improvement in the tracking of the pitch and roll angles. We therefore conclude the hypothesis, that WLS improves the tracking of pitch and roll when producing large yaw moments, to be true.

Finally, from Figure 2 it does become apparent that there still is some small cross coupling between roll and pitch moments and the yaw moment for WLS. The exact cause is beyond the scope of this paper, and may be a topic of future research, but there are possible explanations. For instance, the controller takes into account a linear control effectiveness, while this can be expected to be a quadratic one. Especially for large input changes, as is the case here, some error may be expected. Furthermore, we may consider the fact that for WLS, everything is combined into one cost function. This means that putting more weight on roll and pitch may reduce the error in tracking these angular accelerations, but will never bring it to zero. To improve this, the sequential formulation may be a solution.

	ϕ		θ	
	μ	σ	μ	σ
Pseudo-inverse	12.2	4.8	22.8	9.7
WLS	0.9	0.2	0.5	0.4

Table 1: Mean and standard deviation of the maximum pitch and roll error in degrees.

5 Conclusion

In this paper we have applied the WLS control allocation scheme to incremental nonlinear dynamic inversion control. We propose the following prioritization of controlled forces an moments: first roll and pitch, then thrust, then yaw. This ensures the capability of the vehicle to come back to a stable situation from any attitude. Through an experiment we show that the WLS control allocation with these priorities improves the stability when applying large yaw moments.

The algorithm is readily applicable to other types of MAVs for which priorities in controlled axes can be defined, such as hexacopters, or even hybrid aircraft such as the *Cyclone* [15]. Future research will focus on how constraints in the guidance loop should be taken into account, and how this is affected by limits in the inner loop. Finally, given the strong disturbance rejection properties of the INDI controller, this control allocation scheme is expected to also increase the robustness against faults.

Acknowledgements

The authors would like to thank Anton Naruta for his help with the implementation. This work was supported by the Delphi Consortium.

References

[1] M. W. Oppenheimer, D. B. Doman, and M. A. Bolender. Control allocation for over-actuated systems. In *Mediterranean Conference on Control and Automation*, June 2006. doi: 10.1109/MED.2006.328750.

[2] Abbas Chamseddine, Iman Sadeghzadeh, Youmin Zhang, Didier Theilliol, and Ahmed Khelassi. Control allocation for a modified quadrotor helicopter based on

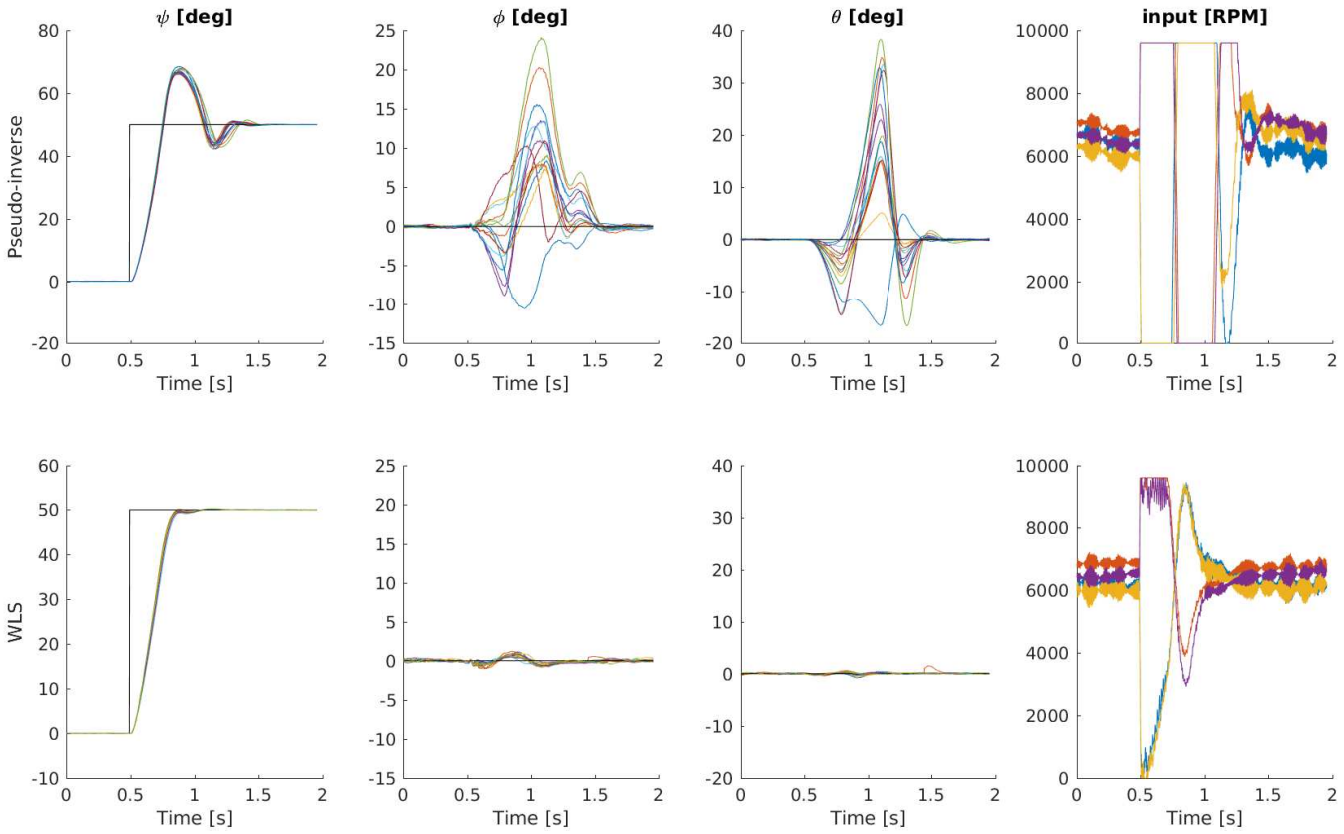


Figure 2: Results of the experiment for the pseudo-inverse (top) and WLS (bottom).

reliability analysis. In *Infotech@Aerospace Conference*, June 2012. doi: 10.2514/6.2012-2511.

[3] Wayne C. Durham. Computationally efficient control allocation. *Journal of Guidance Control and Dynamics*, 24(3):519–524, May 2001. doi: 10.2514/2.4741.

[4] Ewoud J. J. Smeur, Qiping P. Chu, and Guido C. H. E. de Croon. Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Aerial Vehicles. *Journal of Guidance, Control, and Dynamics*, 39(3): 450–461, March 2016. doi: 10.2514/1.G001490.

[5] Ewoud J. J. Smeur, de G.C.H.E de Croon, and Qiping Chu. Gust disturbance alleviation with incremental nonlinear dynamic inversion. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, December 2016. doi: 10.1109/IROS.2016.7759827.

[6] Tor A. Johansen and Thor I. Fossen. Control allocation - a survey. *Automatica*, 49(5):1087–1103, November 2012. doi: 10.1016/j.automatica.2013.01.035.

[7] Ola Härkegård. Efficient Active Set Algorithms for Solving Constrained Least Squares Problems in Aircraft Control Allocation. In *41th IEEE Conference on Decision and Control*, December 2002. doi: 10.1109/CDC.2002.1184694.

[8] P. de Lamberterie, T. Perez, and A. Donaire. A low-complexity flight controller for unmanned aircraft systems with constrained control allocation. In *2011 Australian Control Conference*, pages 284–289, November 2011.

[9] M. Faessler, D. Falanga, and D. Scaramuzza. Thrust mixing, saturation, and body-rate control for accurate aggressive quadrotor flight. *IEEE Robotics and Automation Letters*, 2(2):476–482, April 2017. ISSN 2377-3766. doi: 10.1109/LRA.2016.2640362.

[10] Marc Bodson. Evaluation of optimization methods for control allocation. *Journal of Guidance, Control, and Dynamics*, 25(4):703–711, July 2002. doi: 10.2514/2.4937.

[11] João C. Monteiro, Fernando Lizarralde, and Liu Hsu. Optimal control allocation of quadrotor uavs subject to actuator constraints. In *American Control Conference*, July 2016. doi: 10.1109/ACC.2016.7524963.

[12] Pranay Sinha, Piotr Esden-Tempski, Christopher Forrette, Jeffrey Gibboney, and Gregory Horn. Versatile, modular, extensible vtol aerial platform with autonomous flight mode transitions. In *IEEE Aerospace Conference*, March 2012. doi: 10.1109/AERO.2012.6187313.

[13] John A. M. Petersen and Marc Bodson. Constrained quadratic programming techniques for control allocation. *IEEE Transactions on Control Systems Technology*, 14(1):91–98, January 2006. doi: 10.1109/TCST.2005.860516.

[14] Åke Björck. *Numerical Methods for Least Squares Problems*, pages 187–213. Siam, 1996. ISBN 978-0-89871-360-2. doi: 10.1137/1.9781611971484.

[15] Murat Bronz, Ewoud J J Smeur, Hector Garcia de Marina, and Gautier Hattenberger. Development of a fixed-wing mini uav with transitioning flight capability. In *AIAA Aviation Forum*, June 2017.

Aeroacoustics Investigation on Nano Coaxial Rotor in Hover

Zhen Liu1,* Chen Bu2, Xiangxu Kong3, and Dong Yang3
State Key Laboratory for Strength and Vibration of Mechanical Structures, school of Aerospace Xi'an Jiaotong University, Xi'an, China

ABSTRACT

Aeroacoustics of nano rotor has an impact on the stability and reliability of nano air vehicle. The interference between rotors on aeroacoustics was investigated for nano coaxial counter-rotating rotor. The aerodynamic model of upper rotor of nano rotor was firstly established with sliding mesh technique and multi-blocks method. The unsteady flow field of upper rotor was then studied with LES method and the pressure, the velocity and the vorticity distribution were analyzed. On the basis of the analysis of flow field, the FW-H method was used to study the aeroacoustics of upper rotor. The total sound pressure level and the frequency noise spectrum of the upper rotor were monitored. In order to verify the results of simulation, the test bench for measuring the aeronautics of upper rotor was established. The variation of sound pressure level was measured. The results between the simulation and experiment were compared. It was found that the sound pressure level of monitor point which is near the vortex is high. In the frequency noise spectrum, there is an integer in multiple of the rotation frequency and fundamental frequency, and the peak value decreases with the increase of frequency. The experimental aeroacoustics results match well with that of the computational results so that the computational method is validated. The aeroacoustics of nano coaxial rotor was then studied numerically. When comparing with upper rotor, it was found that the SPL of nano coaxial rotor increased. Flow field analysis showed that the shedding vortices of upper rotor interact the lower rotor resulting the blade-vortex interaction. It is evident that the aeroacoustics was enhanced by the interference of upper rotor and lower rotor.

1 INTRODUCTION

Nano air vehicle (NAV) [1] proposed by Defense Advanced Research Projects Agency (DARPA) in 2005 have attracted more attention in recent years. It is required to fulfill missions in complex and cluttered environments with a hovering capacity that allows for the timely collection of comprehensive intelligence information [2, 3, 4]. Rotary-wing NAV is one kind of NAV driving with rotors with dimension less than 7.5cm in length and payloads of up to 2g and is capable of hovering and flying at a low speed. The nano rotor mainly generates the low-frequency noise due to its low rotational velocity, which increases the difficulty to fly undetected in narrow space. And the NAV structure vibrates with the aeroacoutic load, which might induce the fatigue damage of the structure so that the reliability of the NAV is reduced [5]. For nano coaxial rotor, it generates more noise because of the strong interaction between the upper rotor and the lower rotor. Therefore, it is necessary to investigate the aeroacoustic characteristics of nano coaxial rotor so as to improve the reliability and performance of NAV.

Analytical method and experimental method are usually two methods widely used to study the aeroacoustics of rotor. Kirchhoff method [6] and Ffowcs-Williams/Hawking equation (FW-H) method [7, 8] are two main analytical methods. Kirchhoff meothod can calculate effectivley the nonlinear acoustics including the quadrapole source and is an important method to calculate the high velocity impulse noise of helicopter rotor. However, this method only gives a solution of a total sound pressure but do not isolate the source related to blade thickness, the source linked to the aerodynamic loading and the quadrapole source so that the physical sense of acoustics is not clear. FW-H method usually takes into account a monopole item and a dipole item. But it is difficult to solve the quadrapole item. So it is suitable for the calculation of the source related to blade thickness and the source linked to the aerodynamic loading. FW-H is an appropriate method for the research on the aeroacoustic characteristics if there is no supersonic or transonic phenomenon. As nano coaxial rotor is small and rotates at a low velocity, FW-H method is applicable for nano rotor. In recent years, enormous amount of rotary-wing MAV were developed with the development of the micro air vehicle. Lots of researches were carried out on aeroacoustics of micro rotor. Giorgia etc. [9] applied FW-H method to study the aeroacoutic characteristics of a small propeller with a diameter of 60 cm. Marino [10] also used FW-H method to predict the noise of a propeller of an UAV. However, the aeronautic characteristics of small coaxial rotor are scarcely studied. The influence of rotor interference on aeroacoustics is not taken into account. Some research

*Email address(es): contact_liuz@mail.xjtu.edu.cn

work on full-scalle counter-rotating rotor was carried out. Yi etc. [11] developed a time-spectral RANS method to study the counter-rotating open rotor. The aeroacoustics analysis is carried out using the FW-H equations on the permeable integral surface. Housman etc. [12] used a coupled computational fluid dynamics and computational aeroacoustics numerical approach to study the aeroacoutic characteristics of a contra-rotating open rotor noise. Three-dimensional time-accurate hybrid Reynolds Averaged Navier-Stokes/Large Eddy Simulation (RANS/LES) CFD simulations are performed in the inertial frame. The aeroacoustic analysis is based on a permeable surface FW-H approach evaluated in the frequency domain. However, the inference between upper rotor and lower rotor was not studied. Despite of the fact that the full-scale rotor operates at a high Reynolds number and rotates at a high speed, the research approach can also be taken as a reference to study the nano coaxial rotor. Aeroacoustics experiments can be carried out in an anechoic room or in an open space. The experiment in the anechoic room has the advantage of low environment noise. But it is required that the room size is far larger than the rotor diameter to avoid the circumfluence due to the rotor wake. The experiment in an open space can alleviate the interference but the background noise is large. Giorgia etc. [9] and Marino [10] performed the aeroacoutics experiments of a small propeller in an anechoic room. In our research, an open space with a low-speed wind speed was chosen to carry out the experiment.

In this paper, a three-dimensional time-accurate Large Eddy Simulation (LES) CFD method coupling with FW-H method was used to study the aeroacoutics characteristics of nano coaxial rotor. The pressure and viscous force was taken as the input parameters for FW-H equations. And the quadrapole item was ignored when solving the FW-H equation. And the aeroacoustics experiments for single rotor were carried out to validate the approach. Then the influence of the interference between upper rotor and lower rotor was studied numerically.

2 Methodology and Experimental Setup

2.1 FW-H Equation

Aeroacoustics originates in the Lighthill formula established by Lighthill in 1952. But the most common method for the analysis of aeroacoutics for moving body is the FW-H equations developed by Ffowcs-Williams and Hawking in 1969 [7].

$$\begin{aligned}\frac{1}{c_0^2}\frac{\partial^2}{\partial t^2}\left[pH(f)\right] =& \frac{\partial}{\partial t}\left[\rho_0 V_i \frac{\partial f}{\partial x_i}\delta(f)\right] \\ &- \frac{\partial}{\partial x_i}\left[p_{ij}\frac{\partial f}{\partial x_j}\delta(f)\right] \\ &+ \frac{\partial^2}{\partial x_i \partial x_j}\left[T_{ij}H(f)\right]\end{aligned} \tag{1}$$

In this equation, the impact of the moving body and its boundary layer on the fluid is considered. The first item in the right side of the equation is a surface sound source decided by $\delta(f)$ function on the body surface which is called monopole source or the source related to body thickness. The second item in the right side of the equation is also a surface sound source decided by $\delta(f)$ function on the body surface. But it is a dipole source or a source linked to the aerodynamic loading. The third item is a volume sound source decided by $H(f)$ function distributed outside of the body. It is a quadrapole source which is related to the nonlinear flow. Monopole acoustic source and dipole acoustic source belong to the surface source which predominates in the total aeroacoustics. The quadrapole shall be calculated by integrating the whole flow field which is very difficult. Since the nano coaxial rotor rotates at a very low speed, the aeroacoustic surface source can only be considered.

2.2 Experimental Setup

The purpose of the experiment is to validate the computational method. The experimental setup consists of a rotor, a motor, a speed controller, a DC power, a BK acoustics measurement system and supports. An experimental setup is shown in Figure 1. The upper rotor of the coaxial rotor is tested. A micro brushless rotor with diameter of 13mm and weighted 2.3g was used to drive the rotor. A DC power is used to supply the energy. And the BK acoustics measurement system was used to measure the noise. A pair of nano

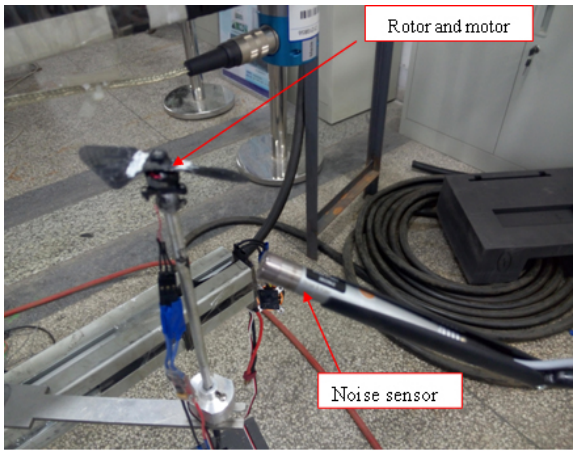


Figure 1: Aeroacoustics test bench for single rotor.

counter-rotating rotors with diameters of 7.5 cm was studied. The upper rotor has a mean chord of 0.33R and a mean twist angle of 17.21° , and the lower rotor has a mean chord of 0.31R and a mean twisted angle of 17.67° as illustrated in Figure **??**. The twist angle and the chord distribution were optimized to ensure the maximum FM at the working RPM during design. Conventional method is not applicable for the improvement of FM. The blade airfoil has a 2% constant thickness with a 5% camber circular arc.

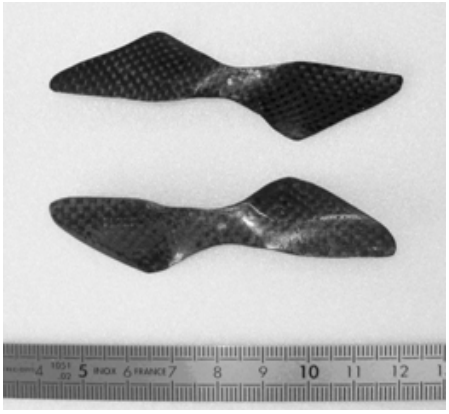


Figure 2: Nano coaxial rotor.

3 Results and discussion

3.1 Aeroacoustics Characteristics of Upper Rotor

Prior to the computation, the meshes of the upper rotor were generated. The multi-block structured mesh was generated. The boundary of the background mesh was formed by a 25.5R-high cylinder with top and bottom radii of 12R and 16R, respectively. In order to calculate the flow field precisely, the boundary method was generated and the thickness of the first layers satisfied with the Yplus in the order of 1. The total number of grid is about 10million.The sliding mesh method was used to describe the motion of the rotor and a three-dimensional time-accurate Large Eddy Simulation (LES) CFD method coupling with FW-H method was used to predict the noise of the upper rotor rotating at a velocity of 6500RPM in static condition. A commercial solver FLUENT was used in this research. The simulations were carried out on the upper rotor at a tip Mach of 0.075. The rotor rotated 1 deg at each physical time step. Rotors rotated more than 10 cycles. The computations were performed on an HP station with 40 Xeon 6290 CPU with 64 GB memory.

Figure 3 shows the vortex contour at the different stations along the blade. According to the theory of vortex and sound, the generation and shedding of the vortices induce pressure fluctuation resulting in the noise. It is found from the figures that trailing edge vortex generates and sheds from the blade. The separation of the boundary layer near the trailing edge induces a large vortex and the vortex sheds due to the instability of the flow. From the blade tip to the blade root, the flow field becomes more complicated and the vorticity magnitude increases. With the analysis of the flow field, the noise monitors were set according to the shedding of the vortex. The acoustics characteristics of the upper rotor were analyzed with FW-H equation method based on the transient analysis of the flow field. From the 4(a), three types of monitor points are presented in order to show the total sound pressure level in the wake. The first type of monitor points are the points located on the centerline of the rotor. The distance of the monitor

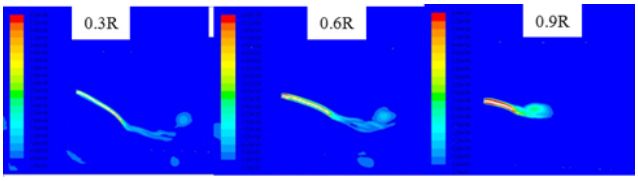
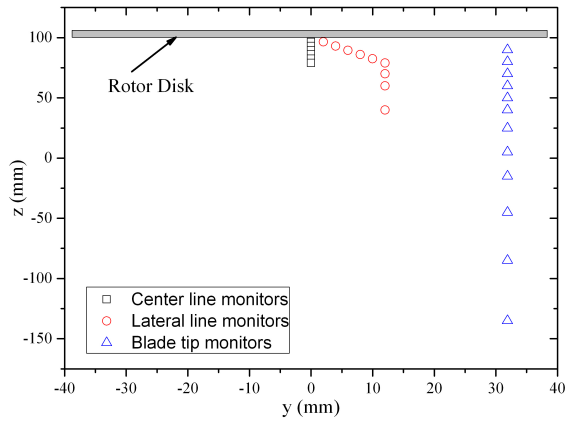


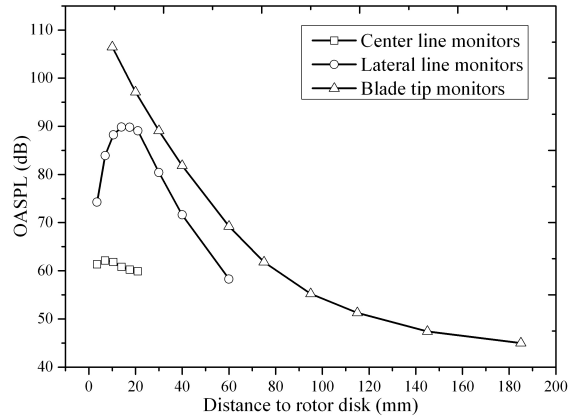
Figure 3: Vorticity contour at different blade station.

point to rotor disk is 3.5 mm, and the distance increases with a increment of 3.5 mm for the following monitor points. The second type of monitor points located in the lateral line of rotor. The first six points has the same vertical distance as the monitor points on the centerline, but the distance in the horizontal direction also increases with an increment of 2 mm. The last three points has the same horizontal distance. The third type of monitor points are generated to capture the tip vortex and locate at the horizontal distance of 0.85 radius of the rotor. Figure 4(b) shows the total sound pressure level at different monitor points. It is found that the OASPL of the monitor points at the blade tip is the highest. The OASPL along the centerline is the lowest. For the first type and the second type of monitor points, the OASPL increases with the distance at the beginning but then decreases with the distance. The OASPL of the monitor points at the blade tip decreases with the distance monotonously. The blade tip vortex, which is one of the main noise source, are very strong at the tip so that the OASPL of the monitor points at the blade tip is high. It is also showed that the OASPL increases from the blade root to the blade tip. Figure 5 shows the total sound pressure level at different monitor points at different radius and angle in the vertical plane. In the rotor disk plane, the OASPL is 95.7dB at 1.2R but it is 81.4R at 1.8R. In the vertical plane, it is found that the OASPL is 85.9dB at the angle of 30 ° but it is 76.5dB at the angle of 60 °. Results illustrate that the OASPL decreases as the radius and the angle increase.

The frequency-analysis could show more information at each monitor points. Therefore, three monitor pints which have the maximum OASPL each type of monitor points were selected to analyze in frequency domain. As the rotational velocity of nano rotor is not too high and the frequency of the rotational noise is in low-frequency range, the cut-off frequency is 4000 Hz. Figure 6 shows the frequency spectrum at different monitor pints. It is found that the peak value for different monitor points is observed at the frequency of 218 Hz which is the fundamental frequency of rotating rotor because its rotational speed is 6500 RPM. From the curve for the monitor point on the centerline, results show that the amplitude fluctuates between -10dB to 20dB and does not change a lot with frequency. However, the amplitude of the acoustic pressure for the monitor point at the lateral line decreases with the increase of the frequency from 0 Hz to 1500 Hz. The same tendency can be found for the monitor point at the blade tip. The amplitude changes dramatically with the frequency be-



(a) Monitor point location



(b) OASPL varying with distance to rotor disk

Figure 4: Total sound pressure level in the wake.

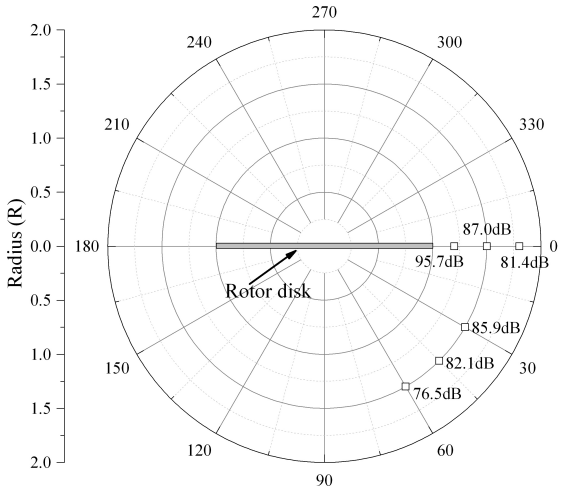


Figure 5: Total sound pressure level at different radius and angle.

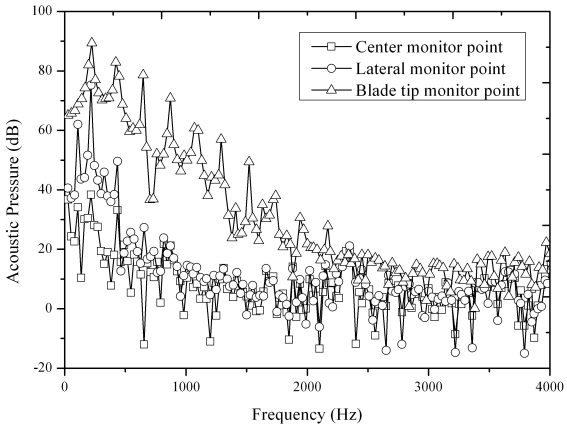


Figure 6: Frequency spectrum of aeronautics of shedding vortex.

fore 2000 Hz. The energy of the noise for single rotor mainly locates in the frequency lower than 2000 Hz. In summary, the aeroacoustic noise of nano rotor is mainly composed of the source related to blade thickness and the source linked to the aerodynamic loading. The peak value of the frequency spectrum curve has relation to shedding vortex and the maximum value is achieved at the fundamental frequency of the rotation. The aeroacoustic sources are surface source. The OASPL decreases with the increase of the distance to rotor disk.

To validate the computational method, the experiments were carried out with the aeroacoustics test bench for single rotor. Figure 7 shows the frequency spectrum of aeronautics at the distance of 20 mm to the blade tip in the wake for both experiment and computation. The frequency spectrum of aeronautics shows the typical characteristics of frequency spectrum for rotational rotor. The maximum peak amplitude locates at the frequency of 226 Hz which close to the fundamental frequency of 218 Hz. It is due to the error of the rotational velocity of the motor. At the frequency multiplication, peak values can be obtained which is due to the shedding of the vortex. When comparing with computational results, more smaller peak values are found. It is caused by the fabrication error of the rotor resulting in the asymmetry of the two blades. At the low frequency the difference between the experiment results and computational results is small, but it is high at high frequency. Figure 8 shows that the OASPL at different monitor points on rotor disk. Results show that the computational results match well with experimental results. In summary, the computational method can predict the aeroacoustics characteristics of nano rotor well.

3.2 *Aeroacoustics Characteristics of Nano Coaxial Rotor*

The distance between upper rotor and lower rotor is 40 mm. The meshes were generated for nano coaxial rotor. The basic parameters of the meshes and the computational method are same to that for single upper rotor. The simulations were

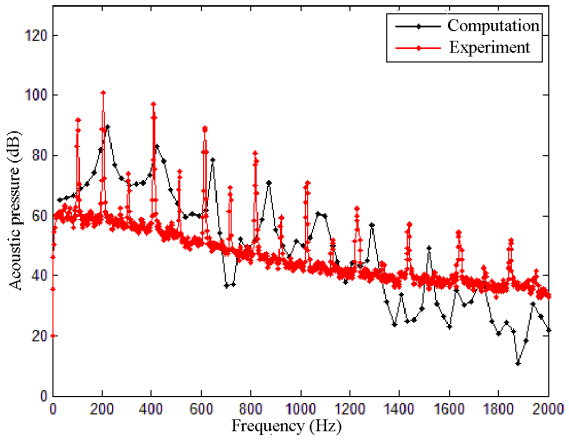


Figure 7: Frequency spectrum of aeronautics at the distance of 20 mm to the blade tip in the wake.

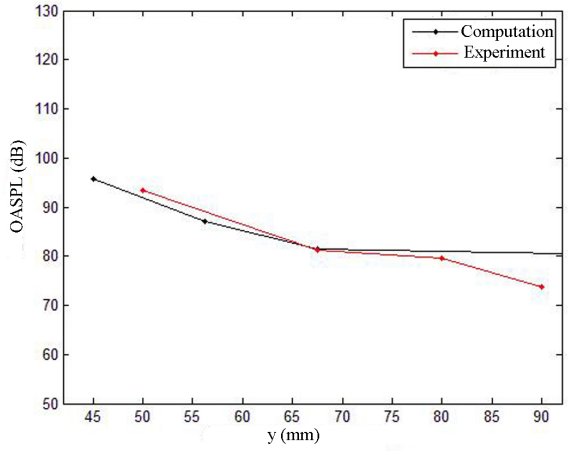


Figure 8: Comparison of OASPL for experiment and computation.

carried out on the both rotors at a tip Mach of 0.075. The rotor rotated 1 degree at each physical time step. Rotors rotated more than 10 cycles.

Figure 9 shows the vorticity magnitude contour at the section of $y = 0$ of the flow field for nano coaxial rotor. The figure illustrates that more vortices are generated between the lower rotor and the upper rotor besides the tip vortex. The strength of the shedding vortex in the wake of the lower rotor is higher than that of the single upper rotor. The vortices in the wake of the upper rotor are stretched due to the interaction of the lower rotor. And the tip vortices of the upper rotor are attract into the upstream flow of the lower rotor. The flow field of the nano coaxial rotor is more complicate than the single upper rotor.

The total sound pressure level was calculated for nano coaxial rotor. Figure 10(a) shows the monitor points in aeroa-

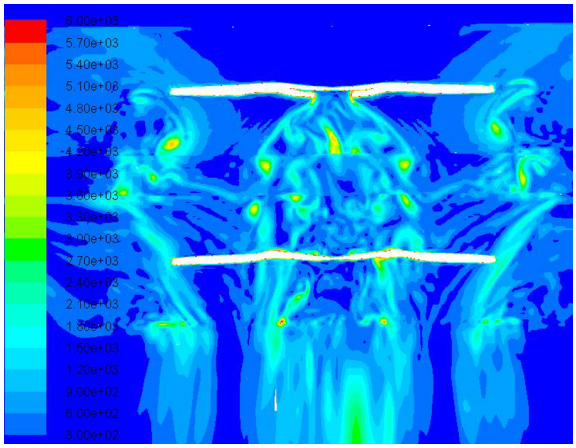
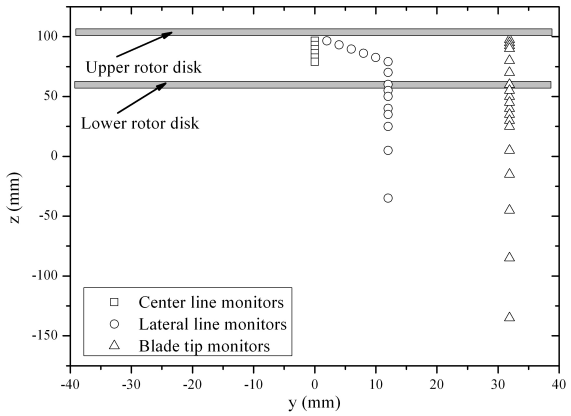


Figure 9: Vorticity magnitude contour of the $y = 0$ section for nano coaxial rotor.

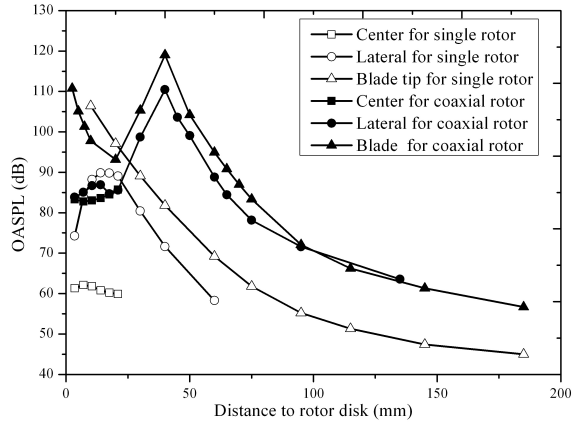
coustic field for nano coaxial rotor. Three types of monitor points were generated as the case for single upper rotor. When comparing the OASPL for single rotor and the coaxial rotor, it is found that the OASPL at the monitor point on the center line for coaxial rotor is higher than that for single rotor as illustrated in Figure 10(b). But for the monitor points on the lateral line and blade tip, one can find that the OASPL for coaxial rotor is lower than that for single rotor when the monitor points locate between the two rotors. But it is far higher than that for single rotor when the monitor points locate below the lower rotor. Results indicates that the interference between upper rotor and lower rotor decreases the total sound pressure level for upper rotor but enhances the total sound pressure level for lower rotor. The total sound pressure level at different radius and angles in vertical plane was calculated. Two radii of 1.5 R and 2.0 Rand four angles of $0°$, $30°$ and $45°$ were chosen to set monitor points as illustrated in Figure 11(a). From Figure 11(b), results show that the OASPL increases with the increase of the angle for both radii of 1.5 R and 2.0 R for the nano coaxial rotor. But the OASPL decreases with the increase of the angle for single rotor. And the OASPL for coaxial rotor is higher than that for single rotor. From 11(a), it can be found that the monitor points get close to the lower rotor with the increase of angle for nano coaxial rotor, which is consider as the reason. The interaction between rotors enhances the OASPL as well.

4 Conclusion

The interference between rotors on aeroacoustics was investigated for nano coaxial counter-rotating rotor. The aeroacoustics characteristics of single upper rotor were firstly studied with transient 3D LES CFD method coupling with FW-H method. And the experiments were carried out to validate the computational method. The total sound pressure level and the frequency noise spectrum of the upper rotor were moni-

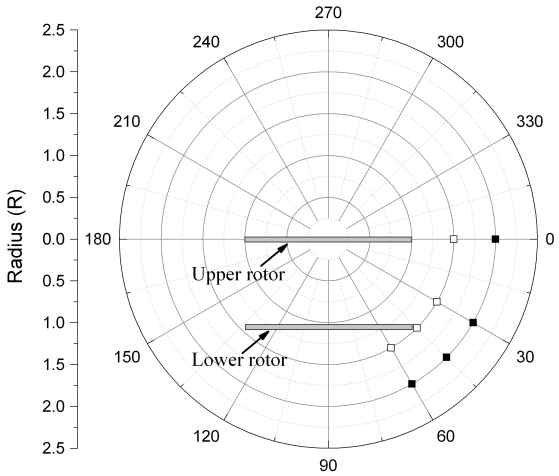


(a) Monitor point location

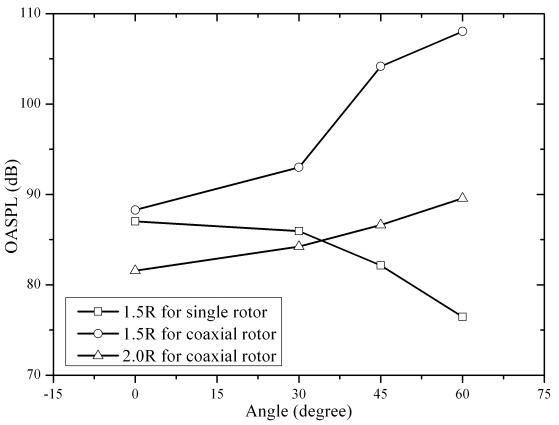


(b) OASPL varying with distance to rotor disk

Figure 10: Total sound pressure level in the wake for nano coaxial rotor.



(a) Monitor point location



(b) OASPL varying with angle and radius

Figure 11: Total sound pressure level varying with the angle and radius.

tored. It is found that the total sound pressure level achieved its maximum value at blade tip. It is considered that the blade tip vortex is one important source for aeroacoustics of rotor. In the frequency noise spectrum, there is an integer in multiple of the rotation frequency and fundamental frequency, and the peak value decreases with the increase of frequency. The experimental aeroacoustics results match well with that of the computational results so that the computational method is validated. The aeroacoustics of nano coaxial rotor was then studied numerically. When comparing with upper rotor, it was found that the SPL of nano coaxial rotor increased. Flow field analysis showed that the shedding vortices of upper rotor interact the lower rotor resulting the blade-vortex interaction. It is evident that the aeroacoustics was enhanced by the interference of upper rotor and lower rotor.

Acknowledgements

The authors would like to thank the support of the National Natural Science Foundation of China (Grant No. 11302164), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20130201120031), and the Fundamental Research Funds for the Central Universities (Grant No. xjj2013031).

References

[1] D.J. Pines. 06-06 proposer information pamphlet (pi) for defense advanced research project agency (darpa) defense sciences office (dso) nano air vehicle (nav) program. In *DARPA DSO*, 2005.

[2] Z. Liu, A. Roberto, J.M. Moschetta, and C. Thipyopas. Experimental and computational evaluation of small micro coaxial rotor in hover. *Journal of Aircraft*, 48(220-229), 1 2011.

[3] S. Sato, M.Drela, J.H. Lang, and D.M. Otten. Design and characterization of hover nano air vehicle propul-

sion system. In *27th AIAA Applied Aerodynamics Conference*, 2009.

[4] H. Youngren, C. Kroninger, M. Chang, and S. Jameson. Low reynolds number testing of the ag38 airfoil for the samarai nano air vehicle. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, 2008.

[5] M.M. Logue and H.M. Atassi. Modeling broadband rotor interaction noise. In *17th AIAA/CEAS Aeroacoustics Conference (32nd AIAA Aeroacoustics Conference)*, 2011.

[6] F. Farassat and M. K. Myers. Extension of kirchhoff's formulation to radiation from moving surface. *Journal of Sound and Vibration*, 123(3):451–460, 1988.

[7] J.E. Ffowcs Williams and D.L. Hawkings. Sound generated by turbulence and surfaces in arbitrary motion. *Philosophical Transactions of the Royal Society*, A264(1151):321–342, 1969.

[8] F. Farassat. Linear acoustic formulas for calculation of rotating blade noise. *AIAA Journal*, 19(9):1122–1130, 1981.

[9] G. Sinibaldi and L. Marino. Experimental analysis on the noise of propellers for small uav. *Applied Acoustics*, 74(1):79–88, 2013.

[10] L. Marino. Experimental analysis of uav-propellers noise. In *16th AIAA/CEAS Aeroacoustics Conference*, 2010.

[11] J.A. Housman and C.C. Kiris. Structured overlapping grid simulations of contra-rotating open rotor noise. In *AIAA 54th Aerospace Sciences Meeting*, 2016.

[12] S. Yi, H. Kwon, D. Im, S. Choi, and D. Lee. Noise characteristics of contra-rotating open rotor with time-spectral cfd and aeroacoustics analysis. In *54th AIAA Aerospace Sciences Meeting*, 2016.

Reducing the noise of Micro–Air Vehicles in hover

R. Serré*, V. Chapin, J.M. Moschetta and H. Fournier
ISAE–Supaero, DAEP, 10 avenue Edouard Belin, 31055 Toulouse, France

ABSTRACT

Micro–Air Vehicles (MAV) are becoming common devices in a wide range of operations while the optimization of their propulsion system is rarely addressed. On the one hand, an aerodynamic optimization would have a straightforward effect on the endurance. On the other hand, an aeroacoustic optimization might increase discretion in military operating conditions, reduce noise pollution in civilian, urban environment and allow sound recordings in dual applications. This contribution aims at presenting a complete methodology for the design of silent and still efficient rotors for MAV, from aerodynamic prediction to aeroacoustic optimization and experimental validation. This approach is suitable for engineering purposes. The aerodynamic and acoustic modeling are described and the optimization procedure is presented. A step–by–step optimization is achieved and measured on an experimental bench suitable for non–anechoic environment. A discussion on the results is proposed. Key parameters on the blade geometry for the reduction of rotor noise are provided at the end of the paper.

1 INTRODUCTION

Designing a silent rotor goes through an aeroacoustic optimization, which implies understanding the aerodynamic phenomenon responsible for noise generation. Predicting the noise generated aerodynamically is relatively straightforward once detailled aerodynamic involved in the propulsion system is available through the use of direct noise computation or hybrid prediction. Aeroacoustic optimization in that framework is possible [1, 2] but demanding in terms of computational cost hence not realistic in an industrial context. Lower–fidelity tools are then needed. Reduction in the rotor noise has received important attention from the early ages of aeroacoustics [3]. It has yielded a lot of informations and materials which allowed development of low–fidelity models of sufficient accuracy. There are identical phenomena that occur in a helicopter rotor and a MAV rotor but the different noise sources do not contribute to the overall noise in the same amount. Detailed analysis of the aerodynamic characteristics has to be specifically dedicated to MAV rotors and low–fidelity models should be re–calibrated or at least carefully selected. For the aerodynamic modeling, a widely spread low–fidelity model is used, based on the Blade Element and Momentum Theory (BEMT) [4]. It is fast, reliable but yields a steady loading on the blades. Acoustic is intrinsically unsteady. Because of the relative motion between the spinning blades and a static observer, acoustic radiation can still be retrieved from a steady loading but it can only be tonal noise as a consequence of a periodic perturbation. As stated by Sinibaldi and Marino [5], the acoustic spectrum radiated by rotors exhibits also a broadband part. Low–fidelity broadband models are then needed in the optimization process in order to get a better description of the acoustic spectrum. The acoustic modeling is realized in two steps: *i)* an integral method based on the Ffowcs Williams and Hawkings [6] (FWH) equation gives the tonal noise radiated by the rotor from the steady loading yielded by the BEMT and *ii)* analytical models estimate the broadband part of the acoustic spectrum based on the work of Roger and Moreau [7]. The optimization process is allowed by several evolutionary algorithms as will be discussed in a future work while results presented hereby are yielded by a combination method, that is a systematic evaluation of the space of parameters. The blade chord and twist laws are parameterized by Bézier curves considering control points in 4 sections a long the blade span giving 8 variables. However, to ensure lift at blade tip reaches zero to yield a minimum induced velocity, the twist at the fourth control point is imposed at zero eventually giving 7 variables. In the combination method, each variables may take 5 values giving 5^7 individual evaluations. A multi–objective selection is applied to express the pareto front according to lower aerodynamic power and lower overall sound pressure level (OASPL). The optimization of the airfoil sections is carried out in a second step through another optimization process. Airfoil shapes are determined using CST parametrization [8] with 12 coefficients. The optimization objective is to maximize the lift-to-drag ratio through NSGA–II evolutionary algorithm with a population of about 100 individuals. The final evaluation is achieved after 55 generations.

2 AERODYNAMIC MODELING

Through a BEMT approach as described by Winarto [4], local distributions of lift and drag and global thrust and torque are retrieved from local lift and drag coefficients of the blade element airfoil sections. As a result, knowledge of the aerodynamic polar of the considered airfoil section is essential to the process. Three strategies may be employed to this end: experimental [9], numerical simulation [10] or numerical modeling (such as panel method in potential flow theory [11]).

*Email address: ronan.serre@isae.fr

The last one is used in the present study for efficiency. Lift and drag coefficients are extracted from Xfoil open–source software by Drela [11], as well as boundary layer data as will be seen in the next section. Figure 1 and figure 2 respectively show lift and drag coefficients prediction by Xfoil compared with experiments from Lyon *et al.* [9] on an E–387 airfoil section at Reynolds number $Re = 100,000$. Xfoil predictions exhibit good agreement with experimental data, a severe drag overstimation at high angle of attack notwithstanding. Figure 3 depicts boundary layer thickness δ on a NACA 0012 at Reynolds numbers $Re = 23,000$ and $Re = 48,000$ and a 6° angle of attack, compared with experiments by Kim *et al.* [12]. The boundary layer behavior experimentally observed is dramatically ignored by Xfoil in the medium chord region which shows a monotonic trend. However, the values does not exhibit too much discrepancy at the trailing–edge region where $x/C \sim 0.04$. Boundary layer data needed for the acoustic modeling is extracted from this region as will be seen in the next section. Xfoil is considered reliable and is used to provide input data for the BEMT approach and broadband noise models.

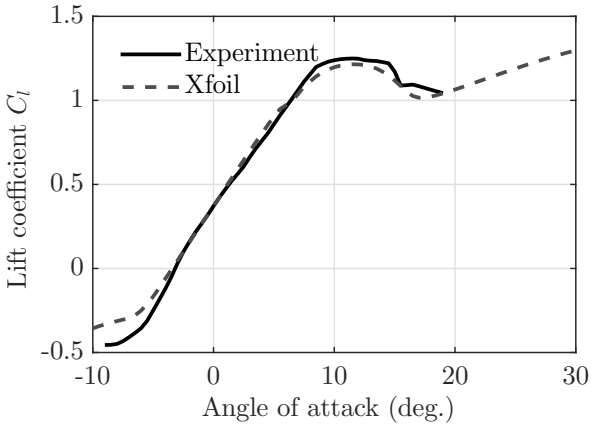


Figure 1: Lift coefficient between Xfoil prediction and experimental work by Lyon *et al.* [9] on an E–387 airfoil section at Reynolds number $Re = 100,000$.

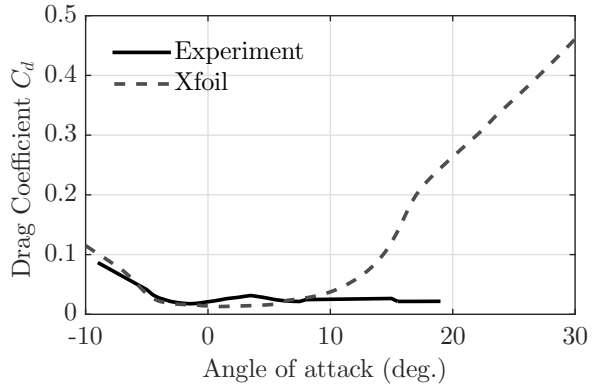


Figure 2: Drag coefficient between Xfoil prediction and experimental work by Lyon *et al.* [9] on an E–387 airfoil section at Reynolds number $Re = 100,000$.

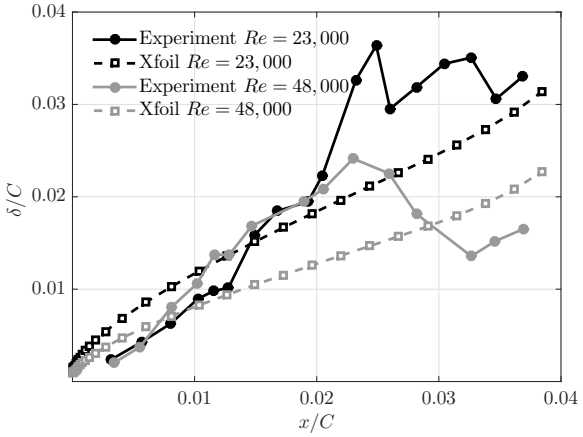


Figure 3: Boundary Layer thickness on a NACA 0012 at Reynolds numbers $Re = 23,000$ and $Re = 48,000$ and a 6° angle of attack between Xfoil prediction and experiments by Kim *et al.* [12].

3 ACOUSTIC MODELING

The FWH equation is implemented in the time domain as expressed by Casalino [13] in the form known as Formulation 1A and applied on the blade surface. Without any fluid volume inside the control surface, the quadrupole term representative of flow non–linearities is neglected but is believed to be of small contribution in this low–Reynolds, low–Mach number regime, typically encountered in MAV rotors [5]. The FWH equation then only resumes to thickness noise and loading noise through surface integrations. The main input parameters are the velocity of the blade element that influence the thickness noise and the force distributions that act on the loading noise. In that steady loading framework, the latter is found to be relatively small without significantly contributing to the overall noise. In this study, the thickness noise is found to be dominant independently of the observer's location. In addition, two sources of broadband noise are considered, based on reference [7]: the scattering of boundary layer waves by the trailing–edge and the ingestion of turbulence at the leading–edge. Roger and Moreau [7] mention a third source of broadband noise, that is the shedding of vortical eddies in the wake. This source will be considered in a future work. The main input for the trailing–edge noise model are a wall–pressure spectrum model as proposed by Kim and George [14] for instance and a spanwise correlation length as modeled by Corcco [15]. The boundary layer data

near the trailing–edge is crucial. This source of broadband noise does not appear to contribute significantly to the overall noise. However, its relevance is supported by the authors to prevent optimization cases where it might overcome tonal noise, as discussed by Pagliaroli *et al.* [16], especially if tonal noise is to be reduced. For the turbulence ingestion noise model, information on impinging turbulence is required. The driving parameters are the cross–correlated upwash velocity fluctuations spectrum such as von Kármán model [17], the mean intensity of the chordwise velocity fluctuations and the Taylor micro–scale as the turbulence length scale. The latter is estimated by the numerical tool from the wake width created at trailing–edge [18] that is believed to impinge the following blade's leading–edge [19]. It is believed by the authors to be the dominant source of noise in MAV rotors [19]. These broadband noise models estimate the noise in the form of a power spectral density, generated in the trailing–edge and the leading–edge regions from boundary layer data and turbulence statistics through a correlation function modified by a Doppler shift imposed by the relative motion between the source and the observer. For the optimization process, only one observer is considered, located $45°$ above the plane of rotation, 1 m away from the center of rotation. Because the noise models exhibit a symetrical behavior with respect to the plane of rotation, selecting an observer position $45°$ above or below that plane of rotation leads to the same conclusions. This location has been chosen as compromise following observation of figure 4 which depicts the overall sound pressure level (OASPL) in dB for the tonal noise, the broadband trailing–edge noise and the total noise of a representative rotor blade, for elevation angles between $\theta = 0°$ (on the axis of rotation) and $\theta = 90°$ (on the plane of rotation). Formulation 1A of the FWH equation gives a singular value on the axis of rotation while the trailing–edge noise model has its singularity on the plane of rotation. These two observation angles should then be avoided.

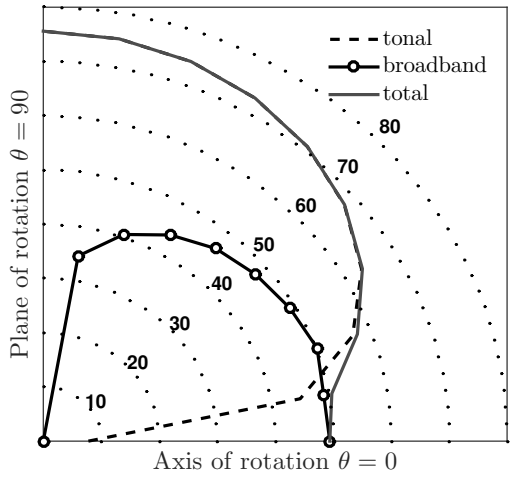


Figure 4: Illustration of the directivity from the noise prediction models. The levels are in dB.

4 Methodology

Relatively low optimization studies on low–Reynolds rotors have been published with regards of the general interest in MAVs and the recent observation that noise from MAVs is generally considered as annoying [20]. Gur and Rosen have proposed a rotor optimization based on aerodynamic efficiency [21]. With aeroacoustic objectives, the reader might refers to Pagano *et al.* [1] and Pednekar *et al.* [2] whose optimization is based on high fidelity numerical simulations. Studies from Wisniewsky *et al.* [22] and Zawodny *et al.* [23] used low fidelity models but at relatively high Reynolds numbers and based on empirical data for symetrical airfoil sections and for that reason theses studies are believed by the authors to lack generality. To demonstrate the feasability of the optimization methodology and to identify the key parameters of the blade geometry allowing noise reduction, a step–by–step optimization of a two–bladed rotor is carried for successive blade geometries:

1. constant chord and constant twist with a NACA 0012 airfoil section
2. same constant chord and optimized twist with a NACA 0012 airfoil section
3. optimized chord and twist with a NACA 0012 airfoil section
4. previous blade geometry with optimized airfoil sections

The successive optimizations occur at iso–thrust, that is to say, the rotational speed is adapted so that the optimized rotors deliver the same thrust, set at 2 N. The optimized geometry is selected to minimize both the aerodynamic power and the OASPL at one specific observer position. At the time the optimizations were carried out, only the trailing–edge noise model was active. The blade geometries are then builded using SLA technolgy on a FormLabs $3D$–printer with a 50 μm vertical resolution for experimental purposes. The maximum radius is the same for all the rotors and is set at $R = 0.0875$ m, imposed by the printing volume allowed by the $3D$–printer.

5 Numerical Results

The successive configurations show an increased twist, along with an increase of the chord for the third optimization. For that optimized rotor, the chord monotonically decreases with the span (figure 5), while the twist is high at the hub, slightly increases at mid–span before reaching a minimal value at the tip (figure 6). The span direction and the

chord are normalized by maximum radius of the rotors. A CAD representation of the four rotors is depicted in figure 7. The fourth rotor is obtained from the third one (with optimized chord and twist) with additional airfoil section optimization. Three airfoil sections at three radial positions are depicted on figure 8. They were obtained by an optimization process as described in the introduction of the present paper to maximize the lift–to–drag ratio on an average in angles of attack at the specified radial positions. The optimized airfoil sections are all thinner than the reference one and cambered as can be expected for low–Reynolds number aerodynamics. The airfoil section near the tip region ($r/R = 1$) exhibits a bump on the suction side, that might indicate an adaptation to separation phenomenon on a specific local Reynolds number. It might be avoided if the airfoil optimization occurs on an average in Reynolds numbers. Figure 9 and 10 show lift and drag coefficients respectively, distributed along the span for the successive blades. The lift coefficient is successively increased with a maximum localized around 75% of the blade radius. The drag coefficient is also increased although less intensively with a maximum value localized around 65% of the blade radius. The lift coefficient is seen to have been multiplied by three while the drag coefficient has been multiplied by two. The gain in aerodynamic efficiency for the successive optimizations yields a diminution of the rotational speed required to deliver the thrust objective set at 2 N (table 1) resulting in a diminution of the blade passing frequency (BPF) (table 2). The tendency of the optimizations to move the BPF towards low frequencies has an effect on the noise reduction for low frequencies are less perceived by human ear. During the optimization process, the sole trailing–edge noise model was active. Figure 11 is presented to assess the ability of the optimization tool to reduce overall noise with the trailing–edge noise model. In figure 11, the blade element contribution to overall noise is shown for the four configurations. For the base configuration, the blade element contribution increases almost linearly toward the tip region according to a Reynolds number effect. The three successive optimizations have a zero twist angle at the tip and it results in a drastically reduced radiated noise near the tip region. The third and fourth optimization cases express a lower radiated noise

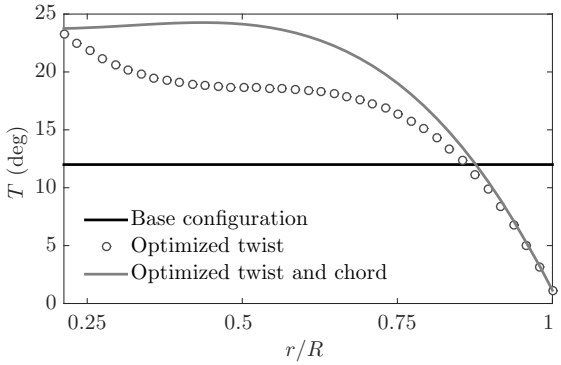


Figure 5: Twist distribution laws of the successive rotors.

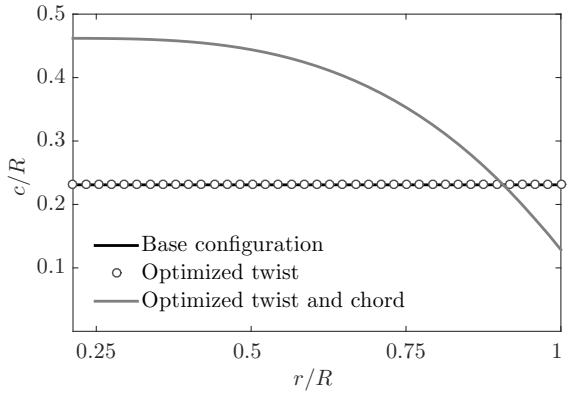


Figure 6: Chord distribution laws of the successive rotors.

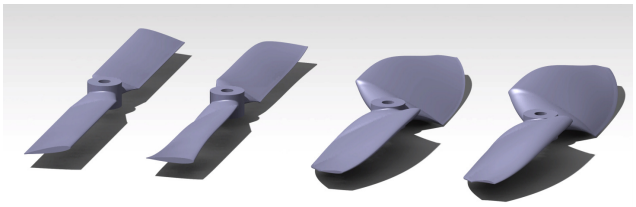


Figure 7: CAD representation of the four rotors considered in the present study. From left to right: initial rotor (base configuration), optimized twist, optimized twist and chord and additional optimized airfoil sections.

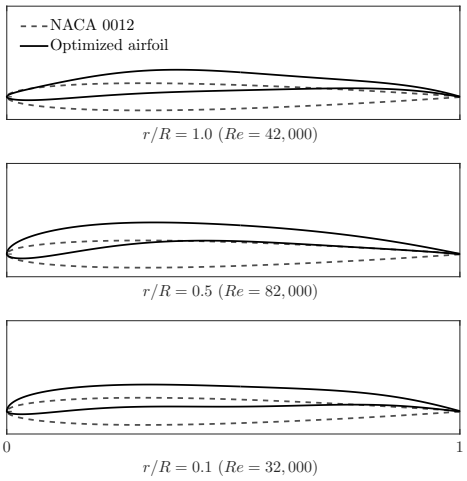


Figure 8: Optimized airfoil sections for the fourth rotor compared with the base configuration (NACA 0012).

	Numerical prediction
Base configuration	9310
Optimized twist	7630
Optimized chord and twist	6010
Optimized airfoil	4880
	Experiment
Base configuration	9800
Optimized twist	8400
Optimized chord and twist	6650
Optimized airfoil	5450

Table 1: Rotational speeds (in rpm) for a 2 N thrust between numerical prediction and experiment for the four successive rotors.

	Numerical prediction
Base configuration	310
Optimized twist	255
Optimized chord and twist	200
Optimized airfoil	165
	Experiment
Base configuration	325
Optimized twist	280
Optimized chord and twist	220
Optimized airfoil	180

Table 2: Blade passing frequency (BPF, in Hz) for a 2 N thrust between numerical prediction and experiment for the four successive rotors.

for each blade element although its chord and twist distribution laws are higher that the second optimization case. The airfoil section optimization increases that tendency. To investigate the noise reduction yielded by the optimization tool for the successive rotors, figures 12 and 13 shows the sound power level predicted by the trailing–edge and the turbulence ingestion noise models, respectively. The sound power level is computed according to ISO $3746 : 1995$ standard in third octave bands for the successive rotors at a 2 N thrust. The important difference in magnitude between the two numerical models is noteworthy. The numerical tool suggests that turbulence ingestion is a more intense source of noise than trailing–edge noise and can overcome the main tonal component at the first BPF. From the two noise models, noise reduction is observed for the successive optimizations. The main tonal noise component that occurs at the first BPF is reduced at each optimization case, up to 25 dB(A) with the fourth rotor as observed in both figures 12 and 13. From the second optimization, the trailing–edge noise is dramatically reduced and the following optimizations increase that tendency (figure 12). The turbulence ingestion noise is also systematically reduced (figure 13).

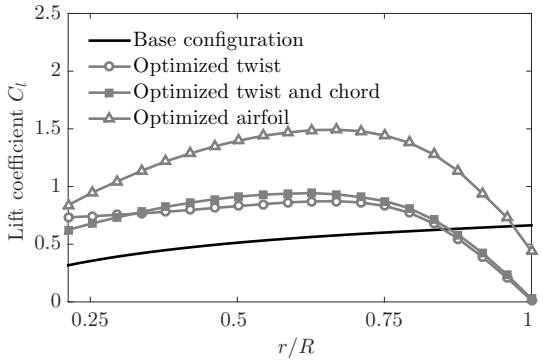


Figure 9: Spanwise lift coefficient distribution of the successive rotors for a 2 N thrust. Numerical prediction.

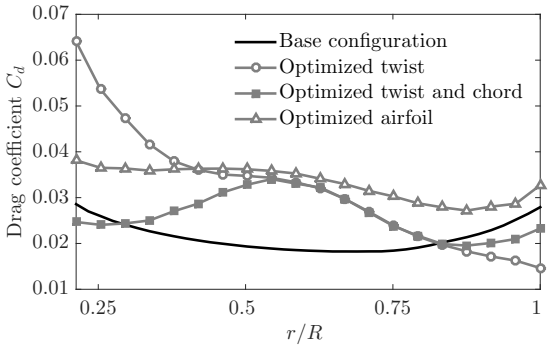


Figure 10: Spanwise drag coefficient distribution of the successive rotors for a 2 N thrust. Numerical prediction.

6 Experiment

The experiment took place in a rectangular room, not acoustically treated, of dimensions $(l_1 \times l_2 \times l_3) = (14.9 \times 4.5 \times 1.8)$ m^3. The aerodynamic forces are retrieved from a five components balance. The sound power level and the total acoustic power are computed according to ISO $3746 : 1995$ standard with five measurement points approximately 1 m around the rotor on Brüel & Kjær $1/2''$ free–field microphones and a Nexus frequency analyzer with a frequency resolution of 3.125 Hz. The distance between the source and the microphones approximately represents 5 rotor diameters. Four of the microphones are on a meridian line parallel to the ground and centered on the axis of rotation. The fifth microphone is located in the plane of rotation. Figure 14 exhibits thrust measurements and numerical predictions for the four successive configurations and several rotational speeds. Measurements and numerical predictions express the same trend, a slight discrepancy observed for the third and fourth optimizations notwithstanding. Such a discrepancy is explained with the last two rotors having a significant mass, resulting in an actual thrust different for the prediction. The mass of the

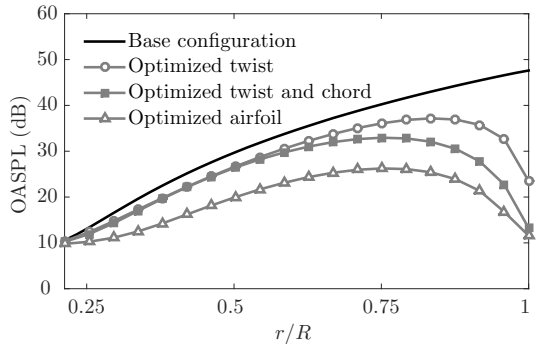


Figure 11: Spanwise blade element contribution to the overall sound pressure level (OASPL) for a 2 N thrust for the successive rotors. Numerical prediction from the trailing–edge noise model.

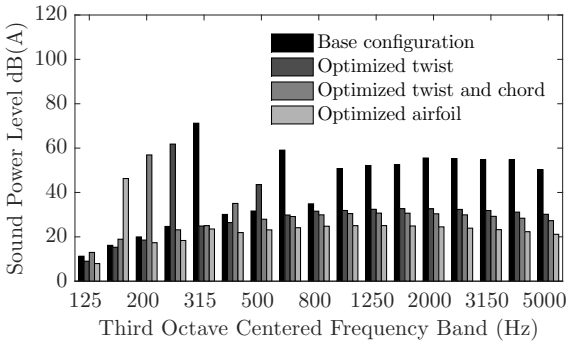


Figure 12: Sound power level of the acoustic spectrum of the successive rotors for a 2 N thrust. Numerical prediction from the trailing–edge noise model.

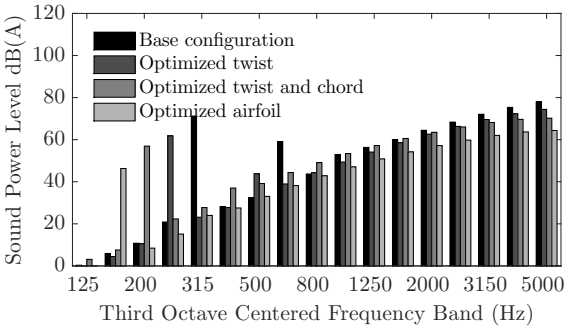


Figure 13: Sound power level of the acoustic spectrum of the successive rotors for a 2 N thrust. Numerical prediction from the turbulence ingestion noise model.

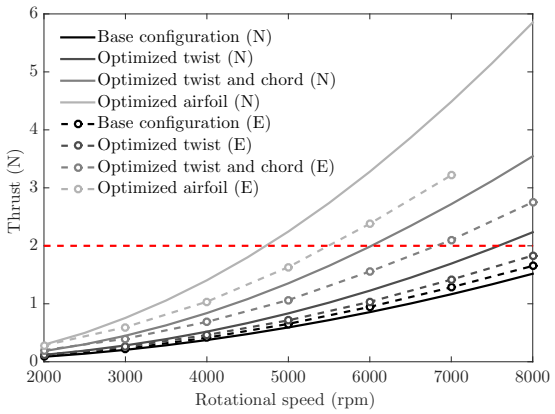


Figure 14: Thrust evolution with rotational speed of the successive rotors from numerical prediction and experiment. The horizontal dash line (red) indicates thrust objective at 2 N. N: numerical predictions. E: experiment.

first two rotors is 8.1 g, while the mass of the last two rotors is 21.0 g representing 4% and 10% of the thrust objective, respectively. Figure 15 shows the sound power level computed according to ISO 3746 : 1995 standard in third octave bands for the successive rotors at a 2 N thrust from experiment. It can be directly compared with figures 12 and 13. Noise reduction is effectively observed, although less than the noise reduction observed from numerical predictions (figures 12 and 13). In the experiment, the main tonal component at the first BPF is reduced by a maximum of 15 dB(A) between the base configuration and the fourth rotor, where the numerical tool predicted a noise reduction by 25 dB(A). Noise reduction occurs in every frequency bands whereas the numerical tool predicted higher low frequencies. Comparing figure 15 with figures 12 and 13 suggests evidences that turbulence ingestion noise might be the dominant source of broadband noise. A slight overestimation by the numerical tool at highest frequencies is however to be expected.

7 Results and Discussion

It appears more clearly on figure 16 why turbulence ingestion noise is believed to be the dominant source of noise in MAV rotors. Figure 16 shows the sound power level computed according to ISO 3746 : 1995 standard in third octave bands for the final optimized rotor at a 2 N thrust from measurements and numerical predictions (trailing–edge and turbulence ingestion noise models). The trailing–edge noise model predicts sound power levels than do not reach the sound power levels observed in the experiment. On the contrary, the turbulence ingestion noise model seems able to predict accurately the broadband components of the sound power spectrum. The experimental data is slightly higher than the numerical predictions but it is reminded that the rotational speed needed to reach the thrust objective is higher in the experiment than in the numerical tool. The exceeding sound power levels seen from the experiments are sub–harmonics

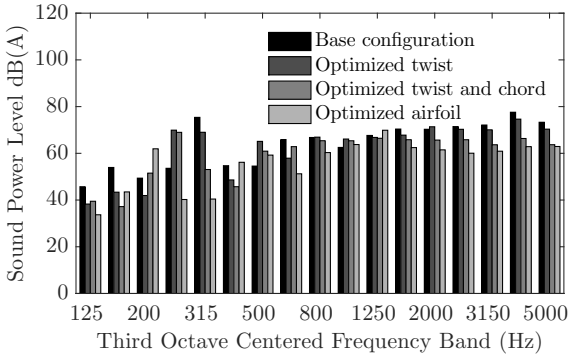


Figure 15: Sound power level of the acoustic spectrum of the successive rotors for a 2 N thrust. Experiment.

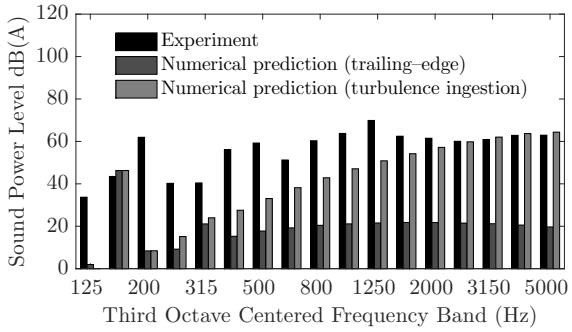


Figure 16: Sound power level of the acoustic spectrum of the final optimized rotor for a 2 N thrust.

tonal peaks that are not retrieved from the tonal noise model in the numerical tool as a consequence of the steady aerodynamic input data. These peaks, as well as the main tonal component at the first BPF that are higher in the experiments are a consequence of unsteady loading occuring during the experiment and may more specifically be a consequence of installation effects. The experimental test bench holds the rotor in such a way that its axis of rotation is parallel to the ground. As a consequence, a stand that includes the aerodynamic balance is mounted vertically, behind the rotor and it might yield additional noise radiation at the BPFs. Moreover, the motor radiates its own noise that has not been identified by the authors. As long as these additional sources of noise are not isolated, a straightforward identification of the sources of noise in the rotor can not be carried out from a typical narrow–band frequency spectrum. Eventually, the following tables exhibit comparison between numerical predictions and experiment on the aerodynamic power (table 3) and on the total acoustic power (table 4). The aerodynamic power is underestimated by the numerical tool by almost 6 W but the power reduction is higher in the experiment (table 3). The total acoustic power is underestimated by the numerical tool with the trailing–edge noise model but is efficiently predicted by the numerical tool with the turbulence ingestion noise model, a slight underestimation for the final configuration notwithstanding. As a result, the reduction in the total acoustic power is amplified by the numerical (table 4).

	Numerical prediction
Base configuration	19.62
Optimized twist	17.18
Optimized chord and twist	17.87
Optimized airfoil	16.87
	Experiment
Base configuration	25.21
Optimized twist	22.12
Optimized chord and twist	23.14
Optimized airfoil	21.27

Table 3: Aerodynamic power in Watts for the four successive rotors for a 2 N thrust.

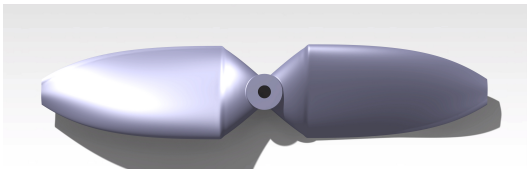
	NTE	NTI	EXP
Base configuration	72.0	85.0	83.3
Optimized twist	61.9	81.2	81.3
Optimized chord and twist	57.0	77.1	76.6
Optimized airfoil	46.6	71.1	74.5

Table 4: Total acoustic power in dB(A)for the four successive rotors for a 2 N thrust. *NTE:* numerical prediction from the trailing–edge noise model. *NTI:* numerical prediction from the turbulence ingestion noise model. *EXP:* experiment.

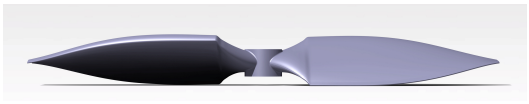
The general trend of the optimization process as shown in tables 3 and 4 is promising: a reduction by 9 dB(A) in the total acoustic power reduction is experimentally observed together with a reduction by 4 W in the aerodynamic power and that is achieved at a minimum cost thank to the numerical tool. Closer views of the most efficient rotor of the successive configurations are shown in figure 17.

8 Conclusion

The successive optimizations presented in this study allow to draw the following conclusions: adapting the twist increases the lift coefficient but more severely increases the drag coefficient as well. Adapting both chord and twist does not affect the lift coefficient but decreases significantly the drag coefficient. Adapting the airfoil section increases again the lift coefficient but with a slight increase in drag coefficient. On the acoustic reduction, the main effect of the optimizations is to provide higher aerodynamic efficiency that allow to decrease the rotational speed which has two effects: *i)* to lower the main frequency of the tonal noise and *ii)* to weaken the intensity of the turbulent eddies that create turbulence ingestion noise. The effect is a direct reduction in



(a) Top view.



(b) Front view.



(c) Side view.

Figure 17: CAD representation of the optimized rotor. It radiates 10 dB(A) less and consummes 4 W less for the same thrust production.

the radiated acoustic energy. The turbulence ingestion is considered the dominant source of broadband noise in MAV rotors. As it is generated in the vicinity of the leading–edge, a biomimetic leading–edge design [24] might help reaching higher levels of noise reduction. Other factors that might also contribute to reduce the noise in MAVs include blade radius and number of blades: increasing the blade radius would increase the aerodynamic efficiency and lower the rotational speed while an odd number of blades is perceived as less annoying as mentionned in a recent psychoacoustic study [20]. This study has contributed to the validation and the demonstration of the efficiency of the low–cost methodology presented in this paper for reducing rotor noise and increasing endurance of Micro–Air Vehicles. The numerical tool and the experimental protocole described in the present paper are suitable for engineering purposes. Reducing the noise from MAVs in hover can then be achieved without expensive means.

Acknowledgements

This study is support by *Direction Générale de l'Armement (DGA)*. Authors thank Sylvain Belliot for the rapid prototyping and Marc C. Jacob for helpful discussions.

References

[1] A. Pagano, M. Barbarino, D. Casalino, and L. Federico. Tonal and broadband noise calculations for aeroacoustic optimization of propeller blades in a pusher configuration. In *15th AIAA/CEAS Aeroacoustics Conference*, number AIAA-2009-3138, 2009.

[2] S. Pednekar, D. Ramaswamy, and R. Mohan. Helicopter rotor noise optimization. In *5th Asian-Australian Rotorcraft Forum*, 2016.

[3] F. Farassat. Theory of noise generation from moving bodies with an application to helicopter rotors. NASA, Technical Report R-451, 1975.

[4] H. Winarto. BEMT algorithm for the prediction of the performance of arbitrary propellers. Melbourne: The Sir lawrence Wackett Centre for Aerospace Design Technology, Royal Melbourne Institute of Technology, 2004.

[5] G. Sinibaldi and L. Marino. Experimental analysis on the noise of propellers for small UAV. *Applied Acoustics*, 74:79–88, 2013.

[6] J. E. Ffowcs Williams and D. L. Hawkings. Sound generation by turbulence and surfaces in arbitrary motion. *Philosophical Transactions of the Royal Society of London*, 264(1151):321–342, 1969.

[7] M. Roger and S. Moreau. Extensions and limitations of analytical airfoil broadband noise models. *International Journal of Aeroacoustics*, 9(3):273–305, 2010.

[8] B. R. Kulfan. Universal parametric geometry representation method. *Journal of Aircraft*, 45(1):142–158, 2008.

[9] C. A. Lyon, A. P. Broeren, P. Giguère, A. Gopalarathnam, and M. S. Selig. Summary of low–speed airfoil data volume 3. SoarTech Publications, Virginia Beach, VA, 1998.

[10] J. Morgado, R. Vizinho, M. A. R. Silvestre, and J. C. Páscoa. XFOIL *vs* CFD performance predictions for high light low Reynolds number airfoils. *Aerospace Science and Technology*, 52:207–214, 2016.

[11] M. Drela and M. B. Giles. Viscous–inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal*, 25(10):1347–1355, 1987.

[12] D. H. Kim, J. H. Yang, J. W. Chang, and J. Chung. Boundary layer and near–wake measurements of NACA 0012 airfoil at low Reynolds numbers. In *47th AIAA Aerospace Sciences Meeting*, number AIAA-2009-1472, 2009.

[13] D. Casalino. An advanced time approach for acoustic analogy predictions. *Journal of Sound and Vibration*, 261:583–612, 2003.

[14] Y. N. Kim and A. R. George. Trailing–edge noise from hovering rotors. *AIAA Journal*, 20(9):1167–1174, 1982.

[15] Y. Rozenberg, M. Roger, and S. Moreau. Rotating blade trailing–edge noise : experimental validation of analytical model. *AIAA Journal*, 48(5):951–962, 2010.

[16] T. Pagliaroli, J. M. Moschetta, E. Bénard, and C. Nana. Noise signature of a MAV rotor in hover. In *49th 3AF international symposium of applied aerodynamics*, 2014.

[17] R. K. Amiet. Acoustic radiation from an airfoil in a turbulent stream. *Journal of Sound and Vibration*, 41(4):407–420, 1975.

[18] T. Fukano, Y. Kodama, and Y. Senoo. Noise generated by low pressure axial flow fans, I: modeling of the turbulent noise. *Journal of Sound and Vibration*, 50(1):63–74, 1977.

[19] R. Serré, N. Gourdain, T. Jardin, A. Sabaté López, V. Sujjur Balaramraja, S. Belliot, M. C. Jacob, and J. M. Moschetta. Aerodynamic and acoustic analysis of an optimized low reynolds number rotor. In *17th International Symposium on Transport Phenomena and Dynamics of Rotating Machinery*, 2017.

[20] N. Kloet, S. Watkins, X. Wang, S. Prudden, R. Clothier, and J. Palmer. Drone on: a preliminary investigation of the acoustic impact of unmanned aircraft systems (UAS). In *24th International Congress on Sound and Vibration*, 2017.

[21] O. Gur and A. Rosen. Optimization of propeller based propulsion system. *Journal of Aircraft*, 46(1):95–106, 2009.

[22] C. F. Wisniewski, A. R. Byerley, W. H. Heiser, K. W. Van Treuren, and W. R. Liller. Designing small propellers for optimum efficiency and low noise footprint. In *33rd AIAA Applied Aerodynamics Conference*, number AIAA-2015-2267, 2015.

[23] N. S. Zawodny, D. Douglas Boyd Jr, and C. L. Burley. Acoustic characterization and prediction of representative, small–scale rotary–wing unmanned aircraft system components. In *72nd AHS Annual Forum*, 2016.

[24] P. Chaitanya, P. Joseph, S. Narayanan, C. Vanderwel, J. Turner, J. W. Kim, and B. Ganapathisubramani. Performance and mechanism of sinusoidal leading edge serrations for the reduction of turbulence–aerofoil interaction noise. *Journal of Fluid Mechanics*, 818:435–464, 2017.

Application of Lattice Boltzmann Method to some challenges related to Micro Air Vehicles

Nicolas Gourdain*, Thierry Jardin, Ronan Serre, Sébastien Prothin and Jean-Marc Moschetta
ISAE-Supaero, 10 av. Edouard Belin, Federal University of Toulouse, France

1 Introduction

Numerical simulation, supported by High Performance Computing (HPC) and experimental validation, led to major breakthroughs in our knowledge of complex physical phenomena and our capability to design innovative technologies. To go beyond the current state of the art, there is still a need to improve our aptitude to deal with complex flow physics, including aerodynamics, aero-acoustics or fluid/structure interactions. Such capabilities are mandatory to address ambitious targets, such as Earth exploration but also the investigations beyond the limit of our planet. Together with the advent of micro-technologies, Micro Air Vehicles (MAVs) recently appeared as a relevant solution for missions of observation and surveillance. MAVs with enhanced endurance and ability to operate in constrained environments would considerably decrease surveillance costs while preserving operators safety in many civilian and military applications. They can also be used in environment where the presence of Humans is not yet possible. However, because of their small dimensions and the intrinsically low Reynolds numbers at which they operate, as well as the difficulty to optimize aerodynamic performance for both forward and hovering flight, current MAVs exhibit relatively low endurance (typically between 15 to 25 minutes in hover) .

Low Reynolds number flows, typical of MAVs ($Re \approx 10^5$), tend to promote flow separations (that decrease efficiency and lift), which are difficult to predict with classical Computational Fluid Dynamics methods, based on a Reynolds Averaged Navier-Stokes (RANS) approach (where all scales of turbulence are modelled). With the increase in computing power, Large Eddy Simulation (LES) emerges as a promising technique to improve the reliability of flow solver predictions [1]. Several works have already show that LES lead to significant improvements both in the understanding of flow physics and performance predictions of rotors [2]. Usually, LES is based on the resolution of the filtered Navier-Stokes equations. While effective, this method require the use of artificial dissipation which limits its accuracy (e.g. transport of turbulence on a long distance, noise predictions, etc.). Lattice-Boltzman Method is a recent and (more and more) popular alternative to such Navier-Stokes flow solvers [3]. Instead of directly solving the Navier-Stokes equations, this method tackles the Boltzmann equation, a statistical equation for the kinetics of gas molecules. Thus, the primitive variables of the LBM represent the statistical particle probability distribution function, to which the usual macroscopic variables pressure and velocity relate as velocity moments, or observables in the sense of statistical mechanics. The particle distribution is a continuous quantity: in contrary to popular believe, the LBM is a continuum method, and not a discrete particle approach. Indeed, the method offers an Eulerian view of the flow and is mesh-based.

To illustrate the advantages and drawbacks of LBM, two applications have been selected in line with MAVs challenges: a rotor operating in-ground effect and new designs to reduce rotor noise. All the LES-LBM simulations have been performed using the open source library Palabos, mainly developed by the University of Geneva.

2 Application to a rotor in-ground effect

This part of the contribution presents numerical investigations undertaken to analyze the turbulent flow produced in the wake of a MAV rotor interacting with the ground [4]. Two configurations are investigated: a free rotor and a shrouded rotor. The Reynolds number based on the chord and tip speed is $Re_{tip} = 0.86 \times 10^5$, which corresponds to a challenging flow where leading edge separations are commonly observed. The numerical simulations are performed with a Reynolds Averaged Navier-Stokes approach and a Large-Eddy Simulation (by means of a Lattice-Boltzmann Method), combined with an immersed boundary approach. The comparison of numerical data with measurements shows that the mean flow and the turbulent shear stresses are accurately predicted close to the ground and in the rotor wake, Fig. 1. However, some discrepancies remain on the prediction of the rotor torque and thrust, mainly due to the difficulty to reproduce the flow near the rotor walls. An analysis is conducted to identify and understand the different sources of turbulent production. The numerical simulations show also that the presence of a shroud contributes, at a given thrust, to reduce the velocity and the turbulent intensity at the ground.

Shrouded rotors usually generate more thrust than equivalent free rotors at a given rotation speed. Therefore, operating at a given total thrust allows reducing

*Email address: nicolas.gourdain@isae.fr

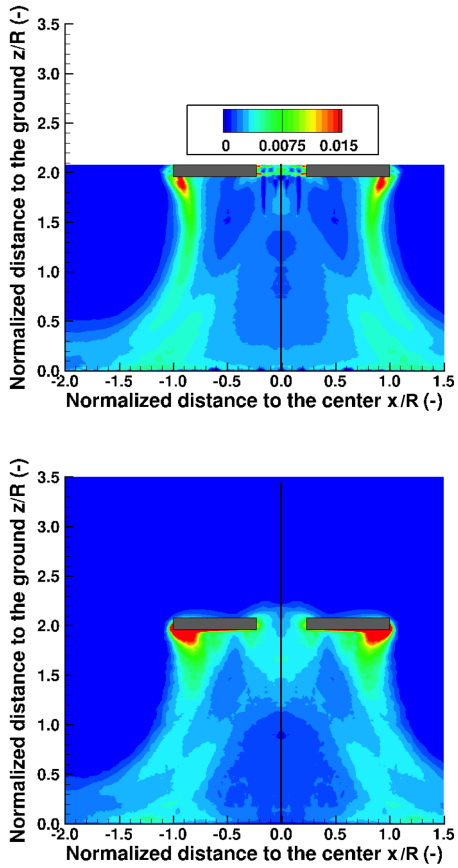


Figure 1: Comparison of the turbulent kinetic energy produced by a MAV rotor in hover, interacting with the ground: (top) measurements and (bottom) LBM prediction.

the part of thrust generated by the rotor itself, hence the rotation speed, thereby reducing the velocity in the rotor wake. In addition, another effect of the shroud is to expand the rotor jet (through a diverging nozzle), which further contributes to decreasing the downward velocity of the rotor wake. All this is conducive to weaker interactions between the rotor and the ground. Ducted and free rotor configurations are compared at a constant rotation rate in Fig. 2. The shroud is responsible for a shift in the location where the rotor wake impacts the ground, that moves from x/R=1.0 (free rotor) to x/R =1.3 (ducted rotor), which corresponds to the effect of the diffuser. No underdeflection is observed at the exit of the diffuser, meaning the boundary layer remains attached despite the adverse pressure gradient. Another effect of the duct is to reduce the thickness of the rotor wake by 20% compared to the free rotor. In the rotor wake ($z/R < 1.5$), the duct reduces the peaks of turbulent kinetic energy (TKE) by 20% and the thickness of the peaks by 50%. The TKE related to the wake mixing in the free rotor case is partially suppressed, which is counterbalanced by a shear layer that develops on the external part of the rotor wake, at the exit of the duct. Close to the ground ($z/R < 0.5$), the turbulent activity remains at the same magnitude order as for the free rotor.

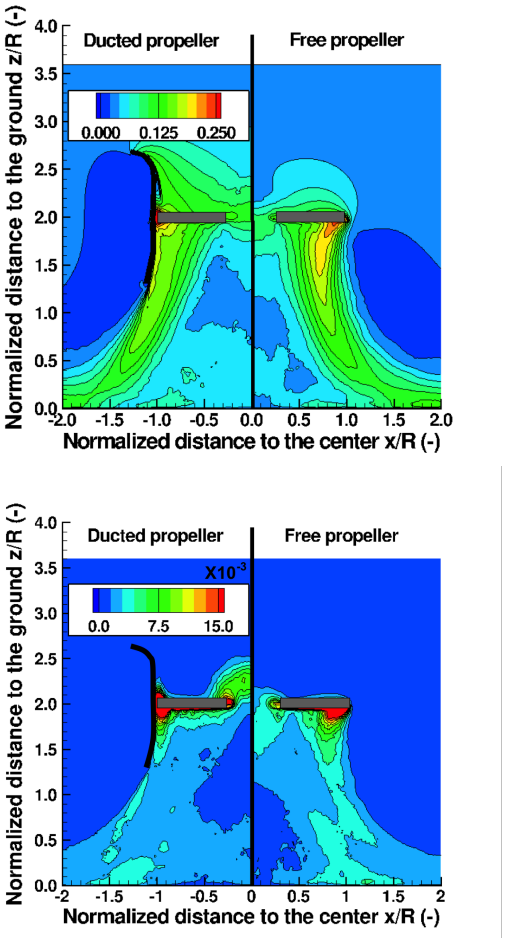


Figure 2: Comparison between a ducted rotor and a free rotor in hover, interacting with the ground: (top) time-averaged velocity flow field and (bottom) time-averaged turbulent kinetic energy flow field.

3 Application to rotor noise

Whether for discretion in military operations or noise pollution in civilian use, noise reduction of MAV is a goal to achieve. Aeroacoustic research has long been focusing on full-scale rotorcrafts. At MAV scales however, the quantification of the numerous sources of noise is not straightforward, as a consequence of the relatively

low Reynolds number that ranges typically from 5,000 to 100,000. Reducing the noise generated aerodynamically in this domain then remains an open topic. This part of the contribution deals with the numerical simulations performed through RANS and LES-LBM to study the flow phenomena that are responsible for the noise generation of a rotor in hover. The initial design is yielded from a low-cost numerical tool developed at ISAE-Supaero based on Blade Element and Momentum Theory (BEMT) for the aerodynamic prediction and formulation of the Ffowcs-Williams and Hawkings (FWH) equation as expressed by [5] coupled with broadband noise models for the acoustic prediction.

Only the global aerodynamic forces are measured with a five components balance. Indeed, the comparison between experimental and numerical predictions is done only for the torque and thrust coefficients, C_Q and C_T, defined as

$$C_T = \frac{T}{\frac{1}{2}\rho(\Omega.R)^2\pi R^2} \; ; \; C_Q = \frac{Q}{\frac{1}{2}\rho(\Omega.R)^2\pi R^3}, \quad (1)$$

with R the radius at the rotor tip. These coefficients are shown in Fig. 3. To check the general shape of the performance curve, another operating point at $\Omega = 314.16 rad.s^{-1}$ (3000 rpm) has been simulated both with URANS and LES-LBM. At 3000 rpm, the accuracy of LES-LBM on thrust is very good (about 1%). Both URANS and BEMT over-predict the thrust coefficient by 15%. When considering the torque coefficient, the order of the methods regarding their accuracy is inverted: LES-LBM, URANS and BEMT over-predicts torque by 50%, 40% and 21%, respectively. At 4950 rpm, similar conclusions can be drawn: the thrust coefficient is under-predicted by 2.5% with LES-LBM, and over-predicted by 14% and 17% by URANS and BEMT, respectively. For the torque, LES-LBM, URANS and BEMT over-predicts it by 29%, 23% and 12%, respectively. It is unclear why three very different numerical methods over-predict the torque (especially the BEMT which neglect 3D effects and predicts fully attached boundary layers). Unfortunately, due to the lack of experimental data, it is not possible at the moment to identify the origin of these discrepancies.

The local thrust coefficient is plotted in Fig. 4 for the three methods. As already shown, BEMT predicts the higher thrust coefficient and LES-LBM the lowest one. From the root to $r/R = 0.4$, the three methods give the same local thrust coefficient. Then, both URANS and LES-LBM predict the same evolution until $r/R = 0.75$ while BEMT already predicts a higher value. All methods show a peak for the thrust coefficient at r/R =0.82 (URANS, LES-LBM) or 0.83 (BEMT). However the values of C_T at the peak are different: C_T=0.19 (LES-LBM), 0.22(URANS) and 0.265 (BEMT). Beyond

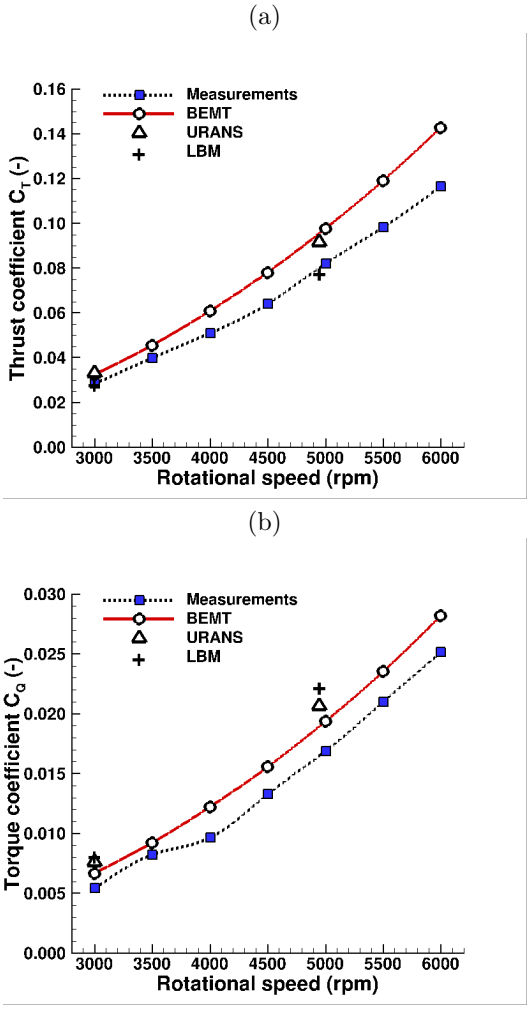


Figure 3: Comparison of global performance: (a) thrust coefficient C_T and (b) torque coefficient C_Q.

$r/R = 0.85$ the value of C_T decreases rapidly. Actually the main conclusion is that the three numerical methods agree reasonably well on a large part of the rotor span, but predict very different behaviour close to the tip.

Three designs are investigated: the reference one, a wavy leading edge design and a design where the blade are shifted along the z-axis. For the three designs, a breakdown of the tip vortex starts close to the trailing edge, due to the recompression. The mixing process between the different vortices produces turbulence that impacts the following blade. A time-averaged flow field, obtained in the reference frame of the rotor, and colored with the normalized fluctuations of pressure $p'/(\frac{1}{2}\rho(\Omega.R)^2)$ is shown in Fig. 5. For the three designs, most of the pressure fluctuations are observed at two locations close to the trailing edge in the vicinity of the

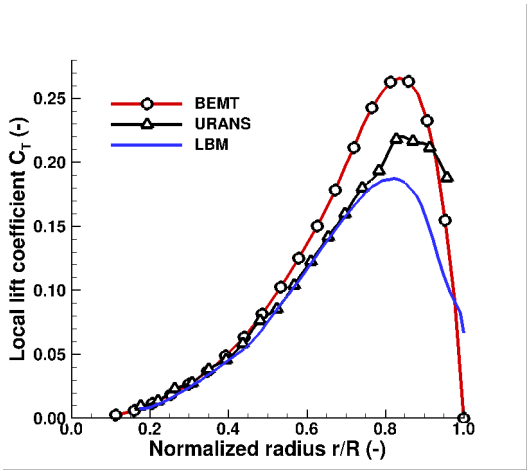


Figure 4: Local thrust coefficient.

rotor tip (tip vortex and suction side separation). The noticeable differences between the three geometries are: 1) for the reference and the wavy leading edge cases, the three blades exhibit the same field of pressure fluctuations, but it is not the case for the shifted blade case (as expected); 2) the source of fluctuations at the rotor tip is reduced with the wavy leading edge; 3) the level of pressure fluctuations in the shifted blade geometry is reduced for two blades (the top-shifted and the middle blade) and is increased for the last one (the bottom-shifted). Compared to the reference geometry, the shifted blade reduces the source at the leading edge but it has a detrimental impact on the trailing edge source.

The global noise of each configuration can be evaluated at a distance of 10 rotor radii, by integrating the pressure signal on all frequencies, which gives a total noise of 1) 64.9dB (reference geometry), 2) 64.1 dB (wavy leading edge) and 3) 68.4 dB (shifted blade).

4 Conclusion

This works shows that LES-LBM has a very good capability to describe the flow around rotors for MAVs applications, both regarding velocity and turbulence. The torque and thrust are also correctly predicted, despite the complexity of the flow (tip vortex, boundary layer separations), with an accuracy close to RANS. Regarding the understanding of flow physics, LES-LBM shows that turbulence is mainly produced by the tip vortex. For a rotor approaching the ground, the slowdown and redirection of the rotor wake is responsible for an intense production of turbulence. Different designs have been investigated to reduce the far-field noise emitted by the rotor. Among the solutions tested, the use of a wavy leading edge is a promising approach to reduce noise. Further investigations should now investigate the poten-

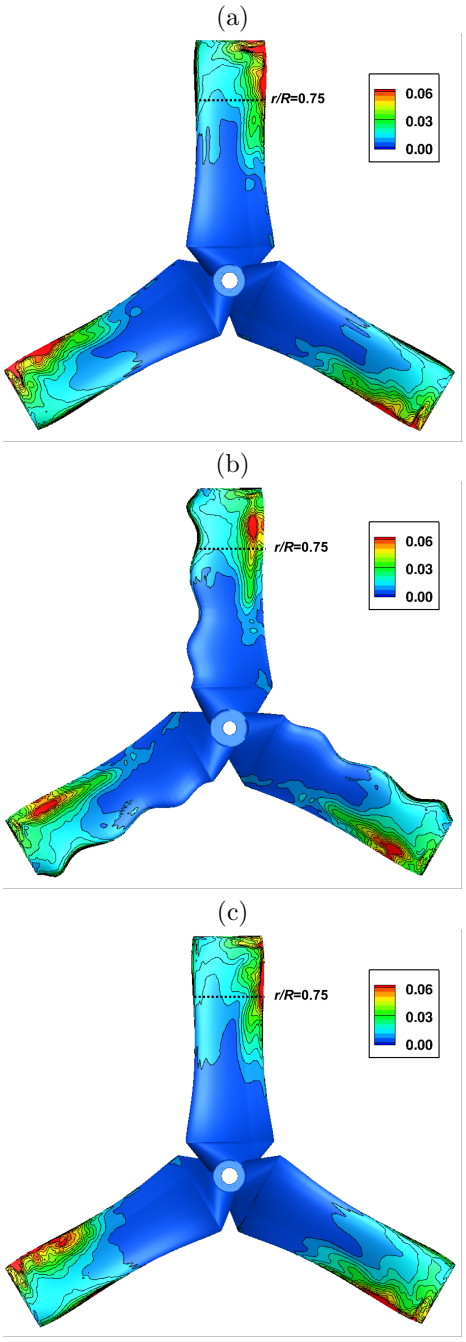


Figure 5: Time-averaged solution on the suction side colored by the normalized pressure fluctations $p'/(\frac{1}{2}\rho(\Omega.R)^2)$: (a) reference geometry, (b) wavy leading edge and (c) shifted blades.

tial of LES-LBM to describe near wall flows and further experimental campaigns are still mandatory to validate the predictions obtained with the LES-LBM technique.

Acknowledgements

Numerical simulations have been performed thanks to the computing center of the Federal University of Toulouse (CALMIP, project p1425) and resources provided by GENCI. These supports are greatly acknowledged. Special thanks to the technical team of DAEP at ISAE-Supaero for the acquisition of experimental data and to Orestis Malaspinas and Jonas Latt, from the University of Geneva, for their support on Palabos. The authors also thank CERFACS, for providing the software used for post-processing and ONERA for the collaborative work on-going around unmanned aerial vehicles.

References

[1] W. Wu and U. Piomelli. Reynolds-averaged and wall-modelled large-eddy simulations of impinging jets with embedded azimuthal vortices. *European J. of Mechanics – B/Fluids*, 55(2), 2016.

[2] R. Ghias, R. Mittal, H. Dong, and T. Lund. Large-eddy simulation of the tip flow of a rotor in hover. In *34th AIAA Fluid Dynamics Conference and Exhibit*, number paper 2004-2432, 2004.

[3] P. Lallemand and L. S. Luo. Theory of the lattice boltzmann method: dispersion, dissipation, isotropy, galilean invariance, and stability. *Phys. Rev. E*, 61(6), 2000.

[4] N. Gourdain, D. Singh, T. Jardin, and S. Prothin. Lattice boltzmann simulation and analysis of the turbulent wake of a mav hovering near the ground. In *European Rotorcraft Forum*, 2016.

[5] F. Farassat. Theory of noise generation from moving bodies with an application to helicopter rotors. In *NASA Tech. Report*, volume R-451, 1975.

Towards a MOMDP model for UAV safe path planning in urban environment

Jean-Alexis Delamer*, Yoko Watanabe†
ONERA - The French Aerospace Lab, 2 Avenue Edouard Belin, 31000 Toulouse
Caroline P. Carvalho Chanel‡
University of Toulouse, ISAE-SUPAERO, 10 Avenue Edouard Belin, 31400 Toulouse

ABSTRACT

This paper tackles a problem of UAV safe path planning in an urban environment in which UAV is at risks of GPS signal occlusion and obstacle collision. The key idea is to perform the UAV path planning along with its navigation and guidance mode planning, where each of these modes uses different sensors whose availability and performance are environment-dependent. A partial knowledge on the environment is supposed to be available in the form of probability maps of obstacles and sensor availabilities. This paper proposes a planner model based on Mixed Observability Markov Decision Process (MOMDP). It allows the planner to propagate such probability map information to the future path for choosing the best action. This paper provides a MOMDP model for the planner with an approximation of the belief states by Mixture of Gaussians.

1 INTRODUCTION

These last years there has been a growing need of UAVs (Unmanned Aerial Vehicles) to accomplish distant missions in urban environments. The feasibility and success of these missions are directly linked to the UAV navigation capacity which relies on onboard sensors (e.g., GPS, vision, . . .). Availability and performance of these navigation sensors may depend on the environment. For example, GPS precision can be degraded due to signal occlusion or multi-path effect in a cluttered environment. Applicability and accuracy of vision-based approaches depend on textures of an image. All of these environment-dependent sensor availabilities and precision affect the navigation performance and thus the safety of the path.

The community has proposed different frameworks and algorithms to solve vehicle safe path planning problem in a cluttered and continuous environment. A method suggested in [1] computes an efficient collision-free path for MAV (Micro Aerial Vehicles) while taking into account uncertainty issued by an onboard camera-based localization. It applies the RRBT (Rapidly-exploring Random Belief Trees, [2]) algorithm. A method proposed by [3] estimates the probability of collision under Gaussian motion and sensor uncertainty. The developed algorithm truncates a state distribution in function of the environment and propagates this truncated state distribution along a path. Other frameworks consider dynamic path being able to self-adapt in function of the events. One could cite the work of [4] that proposes a method based on POMDP (Partially Observable Markov Decision Process) for autonomous vehicles in real situations.

Most of these works do not take into account the variance of sensor availability and precision in function of the environment. Furthermore, the navigation system and its associated uncertainty are often treated in a deterministic manner in the path planner, and a choice of navigation sensors is not included in the elements to be planned.

This paper proposes a planner model which enables the planner to incorporate a priori partial information of the environment-related elements (obstacles, sensor availabilities) in the probabilistic state transition. In this work, such information is given in a form of probability grids which are overlaid on the continuous environment space (Figure 1). Depending on the availability of the sensors, different navigation and guidance modes can be selected, resulting in different localization and path execution error propagation. The planner will be designed to compute a policy which will allow, at each instant, the UAV to choose the best motion-direction and the adapted sensors (i.e. navigation and guidance modes), with regard to the mission efficiency (minimum flight time/length) and safety (minimum collision risk).

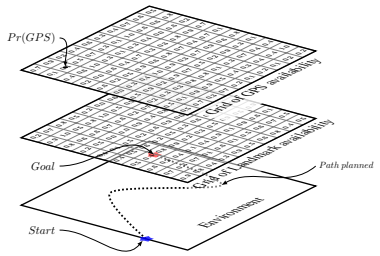


Figure 1: Continuous state space with discretized sensor availabilities maps

*Information Processing and Systems Departement (DTIS), email address: jean-alexis.delamer@onera.fr

†Information Processing and Systems Department (DTIS), email address: yoko.watanabe@onera.fr

‡Design and Control of Aerospace Vehicles Department (DCAS), email address:Caroline.CHANEL@isae.fr

The main contribution of this paper is providing a model which combines the state transition function of the decision process to the localization and path execution error propagation function. Furthermore, the proposed model enables the planner to propagate a priori knowledge on the availability of the navigation sensors on the future path. Considering an uncertainty model propagation, we have chosen to lean our planning model on the Markov Decision Process [5] and more precisely on its extension: Mixed Observability Markov Decision Processes (MOMDP)[6]. The paper also proposes to apply Machine Learning technique to approximate the belief states by a Gaussian mixture.

2 PROBLEM STATEMENT

2.1 System architecture and time differentiation

The architecture of the overall system considered in this problem combines the vehicle Guidance, Navigation and Control (GNC) transition model and the MOMDP planning model (Figure 2). The GNC transition model includes the vehicle motion model, onboard sensor models, and flight control system. The policy given by the MOMDP planner takes as inputs the probability distribution over the current state (called belief state) b_{s_c} and a vector of boolean on sensor availabilities s_v. And it will return an action a which selects the motion direction, and the navigation and guidance modes to realize it. The belief state update is performed with an observation s'_v after GNC state transition. Then this new belief state is used by the policy to define the next action. The MDP

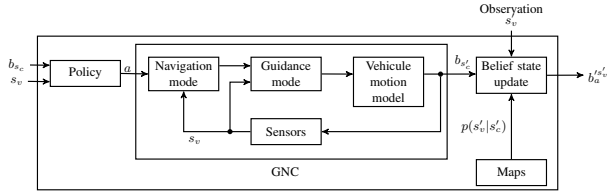


Figure 2: Architecture diagram

formalisms, in their majority and most of the variants, suppose that actions are accomplished instantly. However, in this problem, the actions are durative and the state space is continuous implies that the real state (such as sensor availability) changes during the actions. Therefore, the unit of time of the GNC transition model will be distinguished from that of the MOMDP planning model named *epoch*. The MOMDP planning model works at a lower frequency than the GNC system. Thus an epoch is equivalent to several units of time of the GNC transition model. It is for lowering the complexity of the planning algorithm by reducing the total number of actions to complete the task. The GNC transition unit of time is denoted by k and the planning epoch by t in this paper (Figure 3).

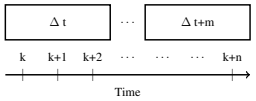


Figure 3: Difference between the units of time

2.2 GNC transition model

This section presents the vehicle GNC model which constructs a state transition model in the planning model.

2.2.1 State transition model

The UAV state $x = \begin{bmatrix}\mathcal{X}^T & \mathcal{V}^T & b_a^T\end{bmatrix}^T$ is defined respectively by its positon, velocity and the accelerometer bias. Then the state transition can be defined such as :

$$x_{k+1} = \Phi x_k + B a_k + v_{k+1} \tag{1}$$

where a_k is the acceleration, $v_{k+1} \sim N(0, Q)$ is the discretized process noise and

$$\Phi = \begin{bmatrix} I & \Delta t I & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, B = \begin{bmatrix} \frac{\Delta t^2}{2} I \\ \Delta t I \\ 0 \end{bmatrix}$$

2.2.2 State estimator

x is estimated by sensor measurements available onboard (Figure 2). The state estimator is based on an EKF (Extended Kalman Filter, [7]) which proceeds in two steps, firstly with the INS prediction then the correction by the sensor measurements, if available. The INS measurement integration enables a high-frequency state estimation, but it suffers from the drift in the estimate. In order to correct such drift, other sensors (e.g. GPS, Vision) are fused with INS through the second correction step.

INS Prediction : The accelerometer measurement a_{IMU_k} is used to propagate the estimated state.

$$a_{\mathrm{IMU}_k} = R_{BI_k}(a_k - g) + b_{a_k} + \xi_{\mathrm{IMU}_k} \tag{2}$$

where R_{BI_k} is a rotation matrix from the inertial to the UAV body frames provided by the INS, g is the gravity vector and $\xi_{\mathrm{IMU}} \sim N(0, R_{\mathrm{IMU}})$ is the IMU measurement noise. According to the process model (1), the estimated state $\hat{x}_k$ is propagated to :

$$\hat{x}^-_{k+1} = \Phi \hat{x}_k + B\left(R^T_{BI_k}\left(a_{\mathrm{IMU}_k} - \hat{b}_{a_k}\right) + g\right) \tag{3}$$

Then the state prediction error is given by :

$$\tilde{x}^-_{k+1} = x_{k+1} - \hat{x}^-_{k+1} = (\Phi - \Delta\Phi^a_k)\,\tilde{x}_k + v_{k+1} - B R^T_{BI_k} \xi_{\mathrm{IMU}_k} \tag{4}$$

where $\Delta\Phi_k^a = BR_{BI_k}^T \begin{bmatrix} 0 & 0 & I \end{bmatrix}$. The associated error covariance is given by :

$$P_{k+1}^- = (\Phi - \Delta\Phi_k^a) P_k (\Phi - \Delta\Phi_k^a)^T + Q + \tilde{R}_{\mathrm{IMU}_k} \quad (5)$$

where $\tilde{R}_{\mathrm{IMU_k}} = BR_{BI_k}^T R_{\mathrm{IMU}} R_{BI_k} B^T$. For simplicity, we can consider the case of $R_{\mathrm{IMU}} = \sigma_{IMU}^2 I$ and hence $\tilde{R}_{\mathrm{IMU}} = BR_{\mathrm{IMU}}B^T$ remains constant for all k.

GPS correction : When GPS is available at t_{k+1}, the predicted state (1) can be corrected by using its position and velocity measurement $z_{\mathrm{GPS}_{k+1}}$.

$$z_{\mathrm{GPS}_{k+1}} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \end{bmatrix} x_{k+1} + \xi_{\mathrm{GPS}_{k+1}} = H_{\mathrm{GPS}} x_{k+1} + \xi_{\mathrm{GPS}_{k+1}}$$

where $\xi_{\mathrm{GPS}} \sim N(0, R_{\mathrm{GPS}})$ is the GPS measurement error. Then, the estimated state is corrected such as :

$$\hat{x}_{k+1} = \hat{x}_{k+1}^- + K_{\mathrm{GPS}_{k+1}} H_{\mathrm{GPS}} \left(z_{\mathrm{GPS}_{k+1}} - H_{\mathrm{GPS}} \hat{x}_{k+1}^- \right) \quad (6)$$

where $K_{\mathrm{GPS}_{k+1}} = P_{k+1}^- H_{\mathrm{GPS}}^T \left(H_{\mathrm{GPS}} P_{k+1}^- H_{\mathrm{GPS}}^T + R_{\mathrm{GPS}} \right)^{-1}$ is a Kalman gain. Then the error estimate and its covariance are given by :

$$\begin{aligned} \tilde{x}_{k+1} &= \left(I - K_{\mathrm{GPS}_{k+1}} H_{\mathrm{GPS}} \right) \tilde{x}_{k+1} - K_{\mathrm{GPS}_{k+1}} \xi_{\mathrm{GPS}_{k+1}} \\ P_{k+1} &= \left(I - K_{\mathrm{GPS}_{k+1}} H_{\mathrm{GPS}} \right) P_{k+1}^- \end{aligned} \quad (7)$$

Landmark by an onboard camera : By image processor, a landmark with a known position $\mathcal{X}_{\mathrm{LM}}$ is visible and detectable at t_{k+1}, it can be used to correct the predicted state (1). By assuming a pin-hole camera model, the pixel-coordinates measurement is given by :

$$z_{\mathrm{LM}_{k+1}} = C \frac{\mathcal{X}_{\mathrm{LM}_{k+1}}^C}{e_3^T \mathcal{X}_{\mathrm{LM}_{k+1}}^C} + \xi_{\mathrm{LM}_{k+1}} = h_{\mathrm{LM}}(x_{k+1}) + \xi_{\mathrm{LM}_{k+1}} \quad (8)$$

where C is a known camera matrix, $\mathcal{X}_{\mathrm{LM}_{k+1}}^C = R_{CB} R_{BI_{k+1}} (\mathcal{X}_{\mathrm{LM}} - \mathcal{X}_{k+1})$ and $\xi_{\mathrm{LM}} \sim N(0, R_{\mathrm{LM}})$ is the landmark image-detection error in pixels. R_{CB} is a camera orientation with respect to the UAV body. An EKF can be applied, and similarly to (7), the resulting estimation error and its covariance matrix are given by :

$$\begin{aligned} \tilde{x}_{k+1} &= \left(I - K_{\mathrm{LM}_{k+1}} H_{\mathrm{LM}_{k+1}} \right) \tilde{x}_{k+1} - K_{\mathrm{LM}_{k+1}} \xi_{\mathrm{LM}_{k+1}} \\ P_{k+1} &= \left(I - K_{\mathrm{LM}_{k+1}} H_{\mathrm{LM}_{k+1}} \right) P_{k+1}^- \end{aligned} \quad (9)$$

where $H_{\mathrm{LM}_{k+1}}$ is a Jacobian matrix of the nonlinear measurement function $h_{\mathrm{LM}}(x_{k+1})$ evaluated at $x_{k+1} = \hat{x}_{k+1}^-$. It should be noted that $H_{\mathrm{LM}_{k+1}}$ thus depends on the predicted state $\hat{x}_{k+1}^-$, while H_{GPS} does not.

INS-only solution : If no correction is made with neither sensors, then state estimate at t_{k+1} is given by:

$$\begin{aligned} \tilde{x}_{k+1} &= \tilde{x}_{k+1}^- \\ P_{k+1} &= P_{k+1}^- \end{aligned} \quad (10)$$

2.2.3 Guidance laws

Given a desired velocity $\mathcal{V}_{\mathrm{ref}}$, the following linear guidance law can be applied to realize it :

$$a_k = \hat{K}_p \mathcal{V}_{\mathrm{ref}} - K_d(\hat{\mathcal{V}}_k - \mathcal{V}_{\mathrm{ref}}) = K_p \mathcal{V}_{\mathrm{ref}} - K_d \hat{\mathcal{V}}_k \quad (11)$$

where K_p,$K_d > O$ are some control gains and $\hat{\mathcal{V}}_k$ is the estimated UAV velocity at instant t_k.

This paper considers two different guidance modes for $\hat{\mathcal{V}}_k$ in (11). The first mode uses the state estimation result from the navigation module (Section 2.2.2), while the second mode uses some sensor measurements directly. The former mode corresponds to a conventional absolute guidance approach such as waypoint tracking, and the latter to a sensor-based relative guidance method such as visual servoing.

Absolute guidance : $\hat{\mathcal{V}}_k = \begin{bmatrix} 0 & I & 0 \end{bmatrix} \hat{x}_k$, where $\hat{x}_k$ is the estimated state from Section 2.2.2. Then, x_{k+1} can be obtained by substituing this guidance law (11) into the discretized process model (1) :

$$x_{k+1} = (\Phi - \Delta\Phi^{\mathcal{V}}) x_k + BK_p \mathcal{V}_{\mathrm{ref}} + \Delta\Phi^{\mathcal{V}} \tilde{x}_k + v_{k+1} \quad (12)$$

where $\Delta\Phi^{\mathcal{V}} = BK_d \begin{bmatrix} 0 & I & 0 \end{bmatrix}$. Hence, the state x_{k+1} follows the normal distribution as below.

$$\begin{aligned} x_{k+1} &\sim N((\Phi - \Delta\Phi^{\mathcal{V}}) x_k + BK_p \mathcal{V}_{\mathrm{ref}}, \Delta\Phi^{\mathcal{V}} P_k \Delta\Phi^{\mathcal{V}^T} + Q) \\ &= N(\bar{x}_{k+1|k}, \tilde{Q}_{k+1}^a) \end{aligned} \quad (13)$$

where the covariance $\tilde{Q}_{k+1}^a$ becomes a function of the estimation error covariance P_k.

Sensor-based relative guidance : In the sensor-based relative guidance mode, $\hat{\mathcal{V}}_k$ in (11) is directly given from some onboard sensors such as optical flow. Let us assume the measurement error $\tilde{\mathcal{V}}_k = (\mathcal{V}_k - \hat{\mathcal{V}}_k) \sim N(0, R_{\mathcal{V}_k})$. Then, similarly to (12),

$$\begin{aligned} x_{k+1} &= (\Phi - \Delta\Phi^{\mathcal{V}}) x_k + BK_p \mathcal{V}_{\mathrm{ref}} + BK_d \tilde{\mathcal{V}}_k + v_{k+1} \\ &\sim N(\bar{x}_{k+1|k}, BK_d R_{\mathcal{V}_k} K_d^T B^T + Q) = N(\bar{x}_{k+1|k}, \tilde{Q}_{k+1}^s) \end{aligned} \quad (14)$$

where the covariance $\tilde{Q}_{k+1}^s$ is now independent from P_k.

2.2.4 State probability density function

Given an initial state $x(t_0) = x_0$ and initial estimation error covariance P_0 such as $\tilde{x}_0 \sim N(0, P_0)$. As defined later in

Section 3.3.2, an action a in the planner model corresponds to a combination of the direction of desired motion $\mathcal{V}_{\text{ref}}$, the navigation mode and the guidance mode. For a given action a, it is possible to obtain the distribution of the next state $x_1 = x(t_1) = x(t_0 + \Delta t)$ knowing x_0.

$$f_X(x_1|x_0) \sim N\left(\bar{x}_{1|0}, \tilde{Q}_1\right) \tag{15}$$

where $\tilde{Q}_1$ is either $\tilde{Q}_1^a$ in (13) or $\tilde{Q}_1^s$ in (14) depending on the selected guidance mode. At the same time, the state estimation error covariance is updated to P_1 by using the selected navigation mode (7, 9 or 10).

Now recall from Section 2.1 that the system's transition model and the planning model do not have the same time unit. It means that a single state transition $s = s_0$ to $s' = s_1$ with an action a in the planner corresponds to several state transitions x_0 to x_n in the system's transition model. Thus, the state transition must continue further up to $n > 1$ with the same action a. The conditional state distribution at t_k knowing x_0 can be obtained sequentially as follows.

$$f_X(x_k|x_0) \sim \int f_X(x_k|x_{k-1}) f_X(x_{k-1}|x_0) dx_{k-1}$$

where $f_X(x_k|x_{k-1}) \sim N(\bar{x}_{k|k-1}, \tilde{Q}_k)$. In parallel, the Kalman filtering process is repeated k times to obtain P_k. When $\tilde{Q}_k$ and P_k do not depend on the state x_{k-1}, the integration above will result in a normal distribution:

$$f_X(x_k|x_0) = N(\bar{x}_{k|0}, \tilde{\Sigma}_k), \quad k > 1 \tag{16}$$

The derivation of the state distribution (16) becomes more complex in a case of having a dependency of $\tilde{Q}_k$ and P_k on the state x_{k-1}. To avoid this complication, $\tilde{Q}_k$ and P_k can be approximated by those evaluated at the expected state $\bar{x}_{k-1|0}$. Then, the state transition function in the planning model can be given by (16) when $k = n$.

Besides, it is hard to know a priori if a selected sensor will be available for the entire action. To simplify we will suppose that if the sensor is available at the end of the action, the sensor was available during the entire action.

In result, the state transition function from the state $s = s_0$ to $s' = s_1$ can be re-wtitten with a notation from the planning model as below.

$$f_S(s' = s_1|s = s_0) = f_X(x_n|x_0) \sim N(\bar{x}_{n|0}, \tilde{\Sigma}_n) = N(\bar{s'}, \Sigma') \tag{17}$$

It is worth emphasizing here that this equation will be the only link to the GNC's transition model with the MOMDP planning model (Section 3.3).

2.3 *Environment Maps*

The planner will use a priori partial knowledge on the environment. This 3D environment map is discretized into cells. Let us denote the c_i, the i-th cell of the map, then a probability that the cell c_i is occupied, i.e., obstacle, is given as $p(Collision|c_i)$. Similarly, sensor availability maps are provided as a set of probabilities that a sensor is available at each cell c_i. They can be generated by considering the corresponding sensor characteristics in relation with the environment (given by the occupancy map). For example, GPS performance suffers from its signal occlusion or multi-path effect due to surrounding obstacles. It is common to measure the performance of GPS by metrics called DOPs (Dilutions of Precision) which corresponds to a standard deviation of the measured position error $\tilde{\mathcal{X}}_{\text{GPS}}$. Figure 4 shows an example of 3D Position DOP (PDOP) map at a given altitude created by a GNSS simulator on an obstacle map of San Diego city. In this paper, this PDOP map is transformed to GPS availability map by setting a maximum allowable position error threshold ϵ. A probability of GPS availability at each cell c_i is calculated by $p(GPS|c_i) = p(\epsilon_{\mathcal{X}_{\text{GPS}}}(c_i) < \epsilon)$.

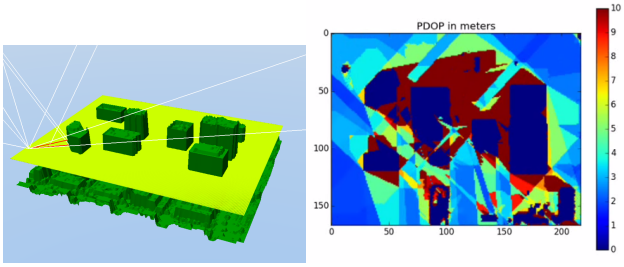


Figure 4: Example of GPS PDOP map

Availability of the navigation mode using landmark detection $p(LM|c_i)$ is obtained in function of a camera's field-of-view and detection performance of an image processor. Likewise, availability of the sensor-based relative guidance (e.g., wall following) is conditioned by a sensing range concerning an object-of-interest (wall). Its availability map as a set of probabilities $p(Wall|c_i)$ is supposed to be given.

3 Planning Model

The objective of this paper is to propose a decision framework for computing the safest and shortest path. To achieve this goal, the transition model must be accurate. That is why the GNC transition model is integrated in the planning model.

3.1 *Why MOMDP ?*

This work is about computing a policy which gives a UAV at each epoch the best action to execute. Therefore, planning can be associated with decision-making which is the cognitive process of choosing an action to perform confronted to a situation. In real life problem, the result of an action is often synonymous of uncertainty that must be considered in the problem. POMDPs and variants provide several frameworks to model sequential decision problem under uncertainty and partial observability. The idea behind the POMDPs is that the state is not known, but several observations are possible for each state with a distinct probability. Then these observations

are used to update its belief state. The Mixed Observability Markov Decision Process (MOMDP) proposed by [8] and [6] is a variant of the POMDPs. The state is not partially observable, but a part of the state is known at each epoch. In this problem, the UAV always knows the current sensor availabilities which can be considered as a part of the state. Consequently, MOMDP can be used in this problem.

3.2 Recall on MOMDPs

The MOMDP explores a particular structure where certain state variables are fully observable. This factored model leads to a significant time gain in policy computation, improving the efficiency of a point-based algorithms. A MOMDP is a tuple $(\mathcal{S}_v, \mathcal{S}_c, A, \mathcal{O}, T, R, b_0, \gamma)$, where:

- $\mathcal{S}_v$ and $\mathcal{S}_c$ are respectively the bounded set of fully observable state variables and the bounded set of partially observable state variables;
- $\mathcal{A}$ is a finite set of actions;
- Ω is a finite set of observation;
- $T(s_v, s_c, a, s'_v, s'_c) \rightarrow [0; 1]$ is a transition function;
- $\mathcal{O}(s', a, o) \rightarrow [0; 1]$ is an observation function such as $p(o_v, o_c|a, s'_v, s'_c)$
- $R : \mathcal{S}_v \times \mathcal{S}_c \times A \rightarrow \mathbb{R}$ is a reward function associated with a state-action pair
- $b_0 = (s^0_v, b_{c,0})$ an initial belief state, where $b_{c,0}$ is the initial probability distribution over the partially observable states, conditionned by s^0_v, the fully observable initial states.
- $\gamma \in [0, 1[$ is the discount factor.

The belief state b is noted by (s_v, b_c), and $\mathcal{B}_c$ is the belief state space of s_c conditioned by s_v : $\mathcal{B}_c(s_v) = \{(s_v, b_c), b_c \in \mathcal{B}_c\}$. $\mathcal{B}_c(s_v)$ is a sub-space of $\mathcal{B}$, such that $\mathcal{B} = \bigcup_{s_v \in \mathcal{S}_v} \mathcal{B}_c(s_v)$.

Solving MOMDPs consists in finding *a set of policies* $\pi_{s_v} : \mathcal{B}_c \rightarrow A$, which maximize the criterion :

$$\pi^*_{s_v} \leftarrow \underset{\pi_{s_v} \in \Pi}{\arg\max} E_{\pi_{s_v}} \left[\sum_{t=0}^{\infty} \gamma^t r((s^t_v, b^t_c), \pi((s^t_v, b^t_c))) \middle| b_0 = (s^0_v, b_{c,0}) \right] \tag{18}$$

For more details about MOMDP, please see [8].

3.3 MOMDP planning model

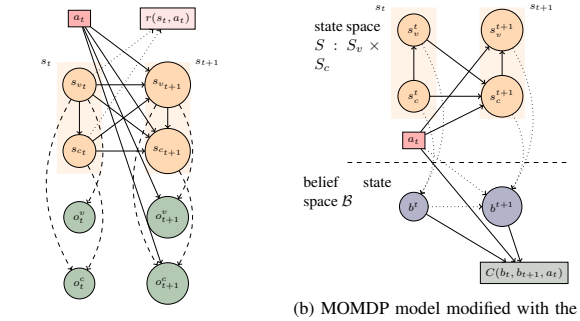
Considering the differences between the problem presented in this paper and a standard MOMDP problem from the litterature, some modifications are made to the MOMDP formalism. Our MOMDP is a tuple $(\mathcal{S}_v, \mathcal{S}_c, \mathcal{A}, \Omega, \mathcal{T}, \mathcal{O}, \mathcal{C}, b_0)$:

- $\mathcal{S}_v$ and $\mathcal{S}_c$ are respectively bounded set of fully observable states and *non* observable states.
- $\mathcal{T}$: The state transition function composed of two functions:
  - $T_{\mathcal{S}_c} : \mathcal{S}_c \times \mathcal{S}_v \times A \times \mathcal{S}_c \rightarrow [0; 1]$ a transition function such as : $T_{\mathcal{S}_c}(s_c, s_v, a, s'_c) = p(s'_c|s_c, s_v, a)$.
  - $T_{\mathcal{S}_v} : \mathcal{S}_v \times \mathcal{S}_c \rightarrow [0; 1]$ a transition function such as : $T_{\mathcal{S}_v}(s'_c, s'_v) = p(s'_v|s'_c)$;
- $\mathcal{O} : \Omega \times \mathcal{S}_c \rightarrow [0; 1]$: observation function such as :

$$O(o, a, s'_c, s'_v) = p(o|s'_c, s'_v, a) = \begin{cases} 1 & o = s'_v \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

- $\mathcal{C} : \mathcal{B} \times \mathcal{B} \times A \rightarrow \mathbb{R}$: the cost function, where $\mathcal{B}$ is the belief state space defined on $\mathcal{S} = \mathcal{S}_v \times \mathcal{S}_c$.

The set of observations Ω is equal to $\mathcal{S}_v$ and consequently the observation function $\mathcal{O}$ is deterministic since $o = s'_v$ in (19). Moreover, the Bayesian dependancy in our model changes from a MOMDP proposed in [8], s'_v depends on s'_c and s'_c depends on s_v and s_c, therefore it depends on the position of the vehicle in the environment (Figure: 5).



(a) Transition model for a MOMDP

(b) MOMDP model modified with the complete belief state space $\mathcal{B}$.

Figure 5: Difference between the transition model

3.3.1 State space of the decisional problem

A state s is composed of two sets : $s_v \in \mathcal{S}_v$ and $s_c \in \mathcal{S}_c$. The state space $|\mathcal{S}| = |\mathcal{S}_v| \times |\mathcal{S}_c|$ represents all the possible states. It must be noted that for our model it is the entire state space $\mathcal{S}$ that is partially observable. More specifically, we define $s_c = x$ (as defined in section 2.2.1), which is defined on a *non observable continuous bounded space*.

s_v is a vector containing the fully observable booleans of sensor availability $[0; 1]$, a boolean on the collision, and P the localization error covariance matrix. In our example, s_v becomes as follows.

$$s_v = \begin{bmatrix} FlagGPS \\ FlagLandmark \\ FlagWallFollowing \\ FlagColision \\ P \end{bmatrix}$$

P is needed by the state probability density function (15). Then, it is necessary to keep P in the state.

3.3.2 Action space of the decisional problem

In contrary to the vehicle state space, the action space is discrete and the action a is defined as a tuple (d, m_n, m_g) composed by three action variables : d, m_n and m_g. $d \in D$ is the desired directions of motion which specifies $\mathcal{V}_{\text{ref}}$ in (11), $m_n \in \mathcal{M}_n$ designates the navigation mode available on the vehicle and $m_g \in \mathcal{M}_g$ designates the guidance mode. In our problem, we define the navigation modes $\mathcal{M}_n = \{INS - only, GPS, Landmark\}$ and the guidance modes $\mathcal{M}_g = \{Waypoint\ tracking, Wall\ following\}$ which corresponds to (10, 7, 9) and (13, 14) respectively.

3.3.3 Observation space and observation function of the decision problem

As explained, a set of observation Ω is considered equal to $\mathcal{S}_v$ and consequently the observation function is deterministic since $p(o|s'_c, s'_v, a) = 1$. In contrary to the standard case of POMDP, the agent does not receive any observation in the classical meaning in our problem. Considering the navigation error propagation, the UAV state s_c is estimated by noisy (and biased for INS) sensor measurements. These measurements are used to estimate the non observable state variables. Unfortunately, we cannot consider these measurements as observations, because it is hard to approach a probabilistic function allowing to anticipate the future measurements. In this sense, s_c is considered as a non observable state variable in the decision problem.

3.3.4 Transition function

The transition function from one state to other is composed of two functions :

- $T_{\mathcal{S}_c}$ a transition function such as :

  $$T_{\mathcal{S}_c}(s_c, s_v, a, s'_c) = f_{s'_c}(s'_c|s_c, s_v, a) \sim N(\bar{s'_c}, \tilde{\Sigma}'(s_v))$$

  As previously developed in (17), the probability distribution of a predicted state s'_c follows a normal distribution $N(\bar{s'_c}, \tilde{\Sigma}'(s_v))$ which is a function of the previous state s_c, the action a. Figure 6a illustrates this function.

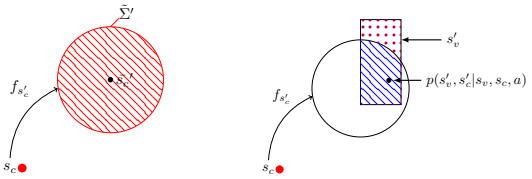
- $T_{\mathcal{S}_v}$ is a transition function such as $T_{\mathcal{S}_v}(s'_c, s'_v) = p(s'_v|s'_c)$. It represents the transition to s'_v and depends on the sensor availability map and therefore depends only on the next state s'_c. Concretely

  $$p(s'_v|s'_c) = \prod_{i=1}^{|\mathcal{S}_v \setminus P|} p(s'_v(i)|s'_c) \tag{20}$$

  where $|\mathcal{S}_v \setminus P|$ is the number of sensors onboard and $s'_v(i)$ is the i-th sensor. Figure 6b illustrates this function.

Then, we define the transition function :

$$\begin{aligned} T(s_v, s'_v, a, s'_c, s_c) &= T_{\mathcal{S}_c}(s_c, s_v, a, s'_c) \times T_{\mathcal{S}_v}(s'_c, s'_v) \\ &= p(s'_v|s'_c) f_{s'_c}(s'_c|s_c, s_v, a) \end{aligned} \tag{21}$$



(a) Propagation of s'_c from s_c with an action a : $T_{\mathcal{S}_c}$

(b) Reduction of the next reachable states in function of s'_v : $T_{\mathcal{S}_v}$

Figure 6: Illustration of the two transition functions : $T_{\mathcal{S}_c}$ and $T_{\mathcal{S}_v}$

3.3.5 Belief state

The belief state condenses all the accumulated information along the path of length N, which is the complete information history h defined by : $h = \{a_0, o_1, a_1, o_1, ..., a_{N-1}, o_N\}$ A belief state b is a probability density function over the possible states at each time step t,

$$b(s^t) = f_s(s = s^t), \forall s^t \in \mathcal{S} \tag{22}$$

computed from the state transition model with selected navigation and guidance modes. This belief state is updated supposing the Bayes rule after each action a done and at each observation o perceived. By factoring the state space according to our model we obtain $b = (s_v, b_{s_c})$, with $b_{s_c} = f_{s_c}(s_c|s_v)$.

3.3.6 Belief state update

As explained, the belief state must be updated after each action associated with the transition function in Section 3.3.4. The update is done by three sub-functions. The first function corresponds to the system's dynamic model which propagates the current belief state b_{s_c} to the futur belief state named $b_{s'_c}$ with a chosen action a :

$$b_{s'_c}(s'_c) = \int f_{s'_c}(s'_c|s_c, s_v, a) b_{s_c}(s_c) ds_c \tag{23}$$

The second is the probability of s'_v that is computed based on $b_{s'_c}$:

$$p(s'_v|b, a) = \sum_{i=0}^{|G|} p(s'_v|s'_c \in c_i) p(s'_c \in c_i|b, a) \tag{24}$$

where c_i corresponding to the ith cell of the sensor availablity map and $|G|$ is the number of cells in the map. The third step corresponds to the "Belief state update" of Figure 2. $b^{'s'_v}_{s'_c,a}$ is computed in function of s'_v, the completely observable state.

$$b^{'s'_v}_{s'_c,a}(s'_c) = \frac{p(s'_v|s'_c)\int f_{s'_c|s_c,s_v,a}(s'_c|s_c,s_v,a)b_{s_c}(s_c)ds_c}{\sum_{i=0}^{|G|} p(s'_v|s'_c \in c_i)p(s'_c \in c_i|b,a)} \tag{25}$$

Therefore the updated belief state is defined as $b^{'s'_v}_a$ and is derived as $b^{'s'_v}_a = (s'_v, b^{'s'_v}_{s'_c,a})$.

3.3.7 Gaussian Mixtures Learning

The representation of a belief state as a Gaussian mixture could simplify the calculation in the planning model. If we consider b^0 as a Gaussian (or a Gaussian mixture), the update of a belief state in Section 3.3.6 raises a problem. Indeed, in (25), the conditional probability $p(s'_v|s'_c)$ (given by the probability grid) make the belief non Gaussian and prevent us to use the initial Gaussian distribution given by (23) (considering b_{s_c} as a gaussian mixture).

A solution proposed in this paper to overcome the problem is to use a machine learning algorithm to approximate the non-Gaussian belief state by a Gaussian mixture model (GMM). More precisely, an "Expectation-Minimization" (EM) algorithm [10] is used to learn the GMM. The idea of the belief state update is unchanged, but the new probability distribution function of the belief state results in GMM learnt with the EM algorithm instead of (25).

Figure 7 illustrates this idea on a given path, the transition between two belief states in this example is just a simple propagation of the Gaussian (or the Gaussian mixture). The belief state update is done by an observation that the GPS is always available. Figure 7a is the GPS availability map used in this example. The graphic on the top of Figure 7b represents the belief states propagated along a given path without any update. The second represents the propagation of each belief states with the update by the observation, approximated by the Gaussian mixture learning.

The effect of the Gaussian mixture learning is more visible nearby an obstacle. The probability on the GPS availability is low, and the original belief states without update are partially in a collision. After using the Gaussian mixture learning, the belief states are updated supposing that an observation, GPS available, is received at each decision step. The belief state is intersected with the GPS availability map, and it results having a smaller state distribution.

3.3.8 Cost function of the model

Our objective is to find the safest and shortest path. To achieve this goal without prioritizing neither the uncertainty nor the length in an artificial way, the cost function is defined based on uncertainty corridor [9]. The corridor is created by a sequence of confidence ellipses, and its volume depends directly on the length of the path and on the dispersion of the uncertainty. Our cost function will be based on this corridor between the two consecutive states $s = (s_v, s_c)$ and $s' = (s'_v, s'_c)$ knowing that the uncertainty is characterized by P which is contained in s_v. Moreover, the cost function includes a direct cost K in case of collisions between states. This cost is needed for the safety of the path planned. If there is an obstacle that intersects the corridor on between two belief states, the second belief state is considered unreachable.

The cost function in MOMDPs needs to be defined over $\mathcal{B}$ instead of $\mathcal{S}$ such as :

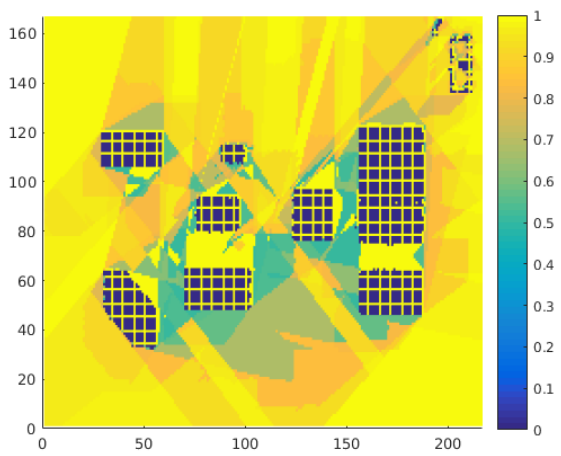
$$\begin{aligned}\mathcal{C}(b, b^{'s'_v}_a) = &\frac{1}{p(s'_v|b,a)}\sum_i p(s'_v|s'_c \in c_i) \\ &\int b_{s_c}\int_{c_i}\mathcal{C}(s,s')f_{s'_c|s_c,s_v,a}(s'_c|s_c,s_v,a)ds'_c ds_c \\ &+ K \times s'_v(Collision)\end{aligned} \tag{26}$$

This cost function is different from the regular reward function of the POMDP. Classically, a reward function corresponding to $R : S \times A \to \mathbb{R}$, where $R(s,a)$ depends directly on the current state s and the action a done. This way, the function $R(b,a)$ become the average of $R(b,a) = \sum_s R(s,a)b(s)$. It is a linear average that approaches the value function of the POMDP with α-vectors based on its PWLC (Piecewise linear convex) property. In our model, the expected cost is no longer a linear average, and thus the value function is not PWLC.

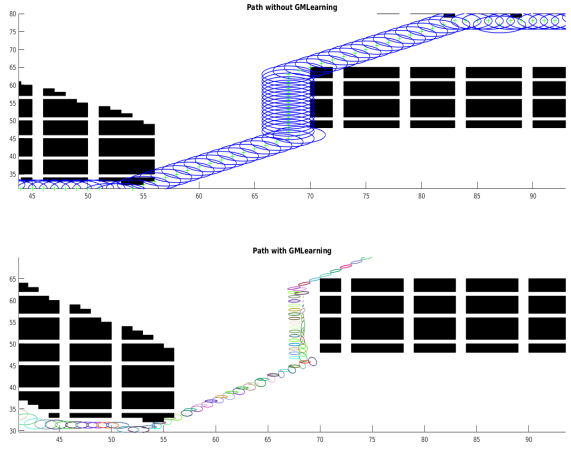
4 Discussion and perspective

There is an important state of the art and the theories behind PWLC in POMDP. It enables to represent the policy by a set of α-vectors which is easier and more intuitive, in contrary to the representation of the policy by a set of all the belief $b \in \mathcal{B}$. Moreover, the theoretical work on the POMDP by [11] ensures that if the value function is PWLC, then value function will be PWLC function at each epoch. The algorithms based on PWLC value function used this particularity to accelerate the computation of the optimum policy. In this work, it is not possible to use these algorithms. One solution is to use algorithms that are not based on PWLC value function. For example, *RTDP-bel* [12] does not work with α-vector, but keeps a hash-table that maps beliefs to values. This algorithm coincides with our problem, but the convergence is not proved. Consequently, the next priority of our work will be to research algorithms that could be adopted or to find a new algorithm that can help to solve the problem.

In this paper, we developed our planning model in perspective to calculate a safe policy in a continuous environment. However, for reducing complexity, our model defines discrete action set which leads to suboptimality in the path planning solution. Therefore in the future, we would like to



(a) Map of probability availability of the GPS



(b) Belief states propagation without (TOP) and with GMM (BOTTOM)

Figure 7: Example of the gaussian mixture learning

incorporate continuous actions, which will reduce the action constraints and then increase the optimality of the path.The planning results depend mainly on the model and the algorithms used. We are also interested in using reinforcement learning to solve the planning model.

References

[1] Markus W. Achtelik, Simon Lynen, Stephan Weiss, Margarita Chli, and Roland Siegwart. Motion- and uncertainty-aware path planning for micro aerial vehicles. *Journal of Field Robotics*, 31(4):676–698, 2014.

[2] Adam Bry and Nicholas Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 723–730. IEEE, 2011.

[3] Sachin Patil, Jur Van Den Berg, and Ron Alterovitz. Estimating probability of collision for safe motion planning under gaussian motion and sensing uncertainty. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3238–3244. IEEE, 2012.

[4] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. Solving continuous pomdps: Value iteration with incremental learning of an efficient space representation. In *In Proc. Int. Conf. on Machine Learning*, 2013.

[5] Richard Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957.

[6] Sylvie CW Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under uncertainty for robotic tasks with mixed observability. *The International Journal of Robotics Research*, 29(8):1053–1068, 2010.

[7] Harold Wayne Sorenson. *Kalman filtering: theory and application*. IEEE, 1985.

[8] Mauricio Araya-López, Vincent Thomas, Olivier Buffet, and François Charpillet. A closer look at momdps. In *Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on*, volume 2, pages 197–204. IEEE, 2010.

[9] Yoko Watanabe, Sylvain Dessus, and Sylvain Fabiani. Safe path planning with localization uncertainty for urban operation of vtol uav. In *AHS Annual Forum*, 2014.

[10] Lei Xu and Michael I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, 8(1):129–151, 1996.

[11] Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

[12] Hector Geffner and Blai Bonet. High-level planning and control with incomplete information using pomdps. In *Proc. Fall AAAI Symposium on Cognitive Robotics*, 1998.

[13] Josep M Porta, Nikos Vlassis, Matthijs TJ Spaan, and Pascal Poupart. Point-based value iteration for continuous pomdps. *Journal of Machine Learning Research*, 7(Nov):2329–2367, 2006.

Cooperative Aerial Payload Transportation Using Two Quadrotors

A. Rajaeizadeh*, A. Naghash, and A. Mohamadifard
Amirkabir University of Technology (Tehran Polytechnic), Faculty of Aerospace Engineering, 424 Hafez Ave., Tehran 15875-4413, Iran

ABSTRACT

In this paper, we consider the problem of controlling multiple quadrotors fastened to a payload and cooperatively transport it in 3 dimensions. We model the quadrotors as a group rigidly attached to a payload. Then we develop the equations of motion of this rigid system. We propose a rigid-body formation system controller based on LQR method as well as a Paparazzi-based guidance scheme for the payload mission trajectory. Also a PD controller is developed and its results are compared with the main controller's results. A simulation study with two quadrotors cooperatively stabilizing, and transporting a payload along two different desired three-dimensional trajectories is presented.

1 Introduction

Safe co-operative transportation of possibly large and bulky payloads is extremely important in various missions such as military operations, Search and Rescue and personal assistant. With recent advancements in relevant technologies and commercially available micro aerial vehicles (MAVs), the problem of autonomous grasping, manipulation, and transportation is advancing to the aerial domain in both theory and experiments. The problem is addressed and formulated either in a way that the payload is connected to the aerial vehicles via flexible cables or gripped to the agents in multiple locations, forming a rigid body in total, the latter of which is the subject of the current job. However these results are based on the common and restricting assumption that the Dynamics of the cable and payloads are ignored and they are considered as a bounded disturbance in the transforming vehicle. Therefore it is challenging to incorporate the effects of a payload fixed to a group of quadrotors which prompts a distributed control policy for each agent while the whole system exhibits a rigid body dynamics. We approach the problem by first developing a model for a single quadrotor and a team of quadrotors rigidly attached to a payload 'Sec. 3'. In 'Sec. 4', we propose a LQR control law for an individual quadrotor. Guidance scheme which consists of a reference generator and a PD controller is issued in 'Sec. 5'. A MATLAB-based simulation is conducted for this problem with figures and discussions found in 'Sec. 6'. The final edition of this paper may contain results from an experimental study which is to be performed in the next step.

*Email address(es): ahmad_rajaiizade@yahoo.com

2 Literature Review

Modeling and control of a quadrotor with a payload which is connected via a flexible cable is addressed in [1]. The work is then extended for multiple quadrotors each carrying a cable which is attached to a single payload [2]. The system modeling is based on Lagrangian mechanics and the flexible cables are modeled as systems of serially-connected links and has been considered in the full dynamic model. We address a different problem as the robots have grasped the payload via rigid connections at multiple locations. The modeling of contact constraints is considerably simpler as issues of form or force closure are not relevant. Additionally, contact conditions do not change in our case. However, the system is statically indeterminate and the coordination of multiple robots is significantly more complex than in the case when the payload is suspended from aerial robots. In particular, as the problem is over-constrained the robots must control to move in directions that are consistent with kinematic constraints. The problem of aerial manipulation using rigid cables (rods and ball joints) is analyzed in [3] with the focus on finding robot configurations that ensure static equilibrium of the payload at a desired pose while respecting constraints on the tension. In the sense that we have access to many rotors to generate the thrust necessary to manipulate payloads, our work is similar to [4], where the authors propose control laws that drive a distributed flight array consisting of many rotors along a desired trajectory. In [5] quadrotors are attached to a flexible net. The fleet is capable of throwing and catching balls with the net. Based on a first-principles model of the net forces, algorithms that generate open-loop trajectories for throwing and catching a ball are introduced. A swarm of quadrotors termed as a flying hand is proposed in [6] which is able to grasp an object where each UAV contributes to the grasping task with a single contact point at the tool-tip and is tele-operated by a human hand whose fingertip motions are tracked. A classification on how the object is carried or grasped is offered in [7] with an elaboration on Aerial Grasp and Manipulation proposing virtual linkage as a new paradigm. According to the offered terminology, our work is termed Aerial Transportation, where

a static aerial object is grabbed by magnetic or gripper-type mechanism. Then, the object is rigidly attached to the mainframe of the UAV. When two or more UAVs are involved, a new UAV arises because the two original mainframes are now rigidly connected to the object. A novel aerial manipulation system is proposed in [8], where a mechanical structure enjoys a number of thrusters and their geometry will be derived from technical optimization problems. The aforementioned problems are defined by taking into consideration the desired actuation forces and torques applied to the end-effector of the system. But the most inspiring work in which the main results of the current research are mostly utilized from is the one proposed in [9] that makes use of a PD controller to stabilize the system. We employed the main results of that work to formulate our control input (force and moments) distribution algorithm for each quadrotor with slight modifications in case of non-symmetrical payload mass distribution, while differing significantly in terms of autopilot controller and guidance scheme.

3 System Modeling

3.1 Generalities

In modeling the rigid-body system following assumptions are made :

- The system is made of two quadrotors that are rigidly attached to a beam shaped payload.
- The center of mass of the whole system coincides the origin of the rigid-body coordinate system.
- Quadrotors and the payload are rigid.
- Propellers are assumed rigid.
- Thrust and drag forces are proportional to the square of propellers speed.
- The system is symmetric in its X axis.

3.2 System Configuration and Coordinate

The NED coordinate system is taken as default coordinate system and it's shown in 'Fig. 1'. The World Frame axes are X, Y and Z pointing downward. We presume the body frame axes are the primary axis of the whole system. Agents have their own coordinate frames and their relative yaw angles which are assumed zero by default.
To derive the rotation matrix from body to the Inertia frame we choose zyx Euler angles rotation. Final matrix is shown as follows:

$$R(\phi, \theta, \psi) = R(x, \phi)R(y, \theta)R(z, \psi) \tag{1}$$

Which will be:

$$R_{IB} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & \psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{2}$$

where $c = cos$ and $s = sin$.

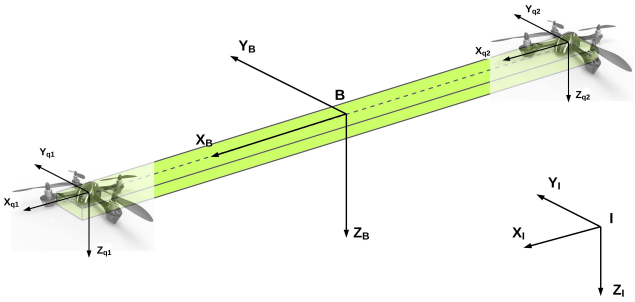


Figure 1: Coordinate Systems.

3.3 Physical Specifications

As the system is composed of 3 separate objects with their own specific physical characteristics, we need to calculate system's overall specifications including mass and moments of inertia. Considering that the agents are identical quadrotors with known physical and geometrical specifications the payload is a rectangular-section beam with a mass of m_p, width of w, height of h and length of L, following data are calculated as:

- Payload's moments of inertia:

$$\begin{aligned} I_{X_p} &= \frac{1}{12} m_p (h^2 + w^2) \\ I_{Y_p} &= \frac{1}{12} m_p (L^2 + h^2) \\ I_{Z_p} &= \frac{1}{12} m_p (L^2 + w^2) \end{aligned} \tag{3}$$

- Position of total center of mass:

$$\begin{aligned} \bar{X} &= \frac{\sum_{i=1}^{3} m_i x_i}{\sum_{i=1}^{3} m_i} \\ \bar{Y} &= \frac{\sum_{i=1}^{3} m_i y_i}{\sum_{i=1}^{3} m_i} \\ \bar{Z} &= \frac{\sum_{i=1}^{3} m_i z_i}{\sum_{i=1}^{3} m_i} \end{aligned} \tag{4}$$

- System's mass and moments of inertia:

$$\begin{aligned} M &= m_1 + m_2 + m_p \\ I_X &= \sum_{i=1}^{3} (I_{x_i} + m(\bar{y_i}^2 + \bar{z_i}^2)) \\ I_Y &= \sum_{i=1}^{3} (I_{y_i} + m(\bar{x_i}^2 + \bar{z_i}^2)) \\ I_Z &= \sum_{i=1}^{3} (I_{z_i} + m(\bar{y_i}^2 + \bar{x_i}^2)) \end{aligned} \tag{5}$$

In which $\bar{x_i} = x_i - \bar{X}$, $\bar{y_i} = y_i - \bar{Y}$ and $\bar{z_i} = z_i - \bar{Z}$.

3.4 Equations of Motion

3.4.1 Single Quad:

First we derive the equations for a single quadrotor. Then these equations will be developed for a multi agent system.

- Rotational Kinematics:

Since the angular rates are related to the body, they should be transformed to Inertia World Frame. Using Euler angles we have:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = R_r \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{6}$$

$$R_r = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \tag{7}$$

- Rotational Dynamics:

Using the rotation matrix and Coriolis Effect, rotational dynamic equation takes the form of:

$$\begin{bmatrix} I_{x,j} & 0 & 0 \\ 0 & I_{y,j} & 0 \\ 0 & 0 & I_{z,j} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{x,j} & 0 & 0 \\ 0 & I_{y,j} & 0 \\ 0 & 0 & I_{z,j} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$= M_{q,j} + M_{Gq,j} \tag{8}$$

where:

$$M_{q,j} = \begin{bmatrix} lb(\Omega_{j,4}^2 - \Omega_{j,2}^2) \\ lb(\Omega_{j,3}^2 - \Omega_{j,1}^2) \\ d(-\Omega_{j,1}^2 + \Omega_{j,2}^2 - \Omega_{j,3}^2 + \Omega_{j,4}^2) \end{bmatrix} \tag{9}$$

$$M_{Gq,j} = J_r \Omega_j \begin{bmatrix} \dot{\theta} \\ -\dot{\phi} \\ 0 \end{bmatrix} \tag{10}$$

$$\Omega_j = (\Omega_{j,1} - \Omega_{j,2} + \Omega_{j,3} - \Omega_{j,4}) \tag{11}$$

- Translational Dynamics:

Applying the rotation matrix we have:

$$m_j \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ m_j g \end{bmatrix} + R_{IB} F_{q,j} \tag{12}$$

$$F_{q,j} = \begin{bmatrix} 0 \\ 0 \\ -b(\Omega_{j,1}^2 + \Omega_{j,2}^2 + \Omega_{j,3}^2 + \Omega_{j,4}^2) \end{bmatrix} \tag{13}$$

3.4.2 The Rigid-body System:

As the whole system is supposed rigid its dynamic model is simpler than a system with cable-suspended load. Considering each agent produces forces and moments in its own frame, we need to develop a relationship between the behavior of the system and the agents. Depending on the configuration, the following formulation was developed in which j, x_j, y_j and ψ_j are the number, location and relative heading of each agent in rigid-body coordinate system.

$$\begin{bmatrix} F_B \\ M_B \end{bmatrix} = \sum_{j=1}^{2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ y_j & cos\psi_j & -sin\psi_j & 0 \\ -x_j & sin\psi_j & cos\psi_j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_{q,j} \\ M_{q,j} \end{bmatrix} \tag{14}$$

Applying the new formulation for the systems force and moments, dynamic of the system will be achieved as follows:

$$\begin{aligned} \ddot{X} &= (sin\phi sin\psi + cos\phi sin\theta cos\psi)\frac{-U_1}{M} \\ \ddot{Y} &= (-sin\phi cos\psi + cos\phi sin\theta sin\psi)\frac{-U_1}{M} \\ \ddot{Z} &= g + (cos\phi cos\theta)\frac{-U_1}{M} \end{aligned} \tag{15}$$

$$\begin{aligned} \ddot{\phi} &= \frac{I_Y - I_Z}{I_X}\dot{\theta}\dot{\psi} - \frac{J_r\Omega\dot{\theta}}{I_X} + \frac{U_2}{I_X} \\ \ddot{\theta} &= \frac{I_Z - I_X}{I_Y}\dot{\phi}\dot{\psi} + \frac{J_r\Omega\dot{\phi}}{I_Y} + \frac{U_3}{I_Y} \\ \ddot{\psi} &= \frac{I_X - I_Y}{I_Z}\dot{\theta}\dot{\phi} + \frac{U_4}{I_Z} \end{aligned} \tag{16}$$

4 Controller Design

4.1 Cooperative Control Law:

As the Agents are rigidly attached to the payload, design of the controller will become a challenge. The first step is to assume the rigid-body system as a single quadrotor with specification of the system. Then we calculate the necessary control commands and try to control this quadrotor. For controlling the attitude of the system two techniques were used, PD and $Tracking\ LQR$. In order to distribute controller commands between agents a comprehensive cooperative control law is developed. Defining [9]:

$$u_F = \frac{1}{M}\left[(m_1 + m_{p1}), 0, 0, 0, (m_2 + m_{p2}), 0, 0, 0\right]^T$$

$$u_{M_x} = \frac{1}{\frac{\omega_{Mxy}}{\omega_F}\sum_{j=1}^{2} y_j^2 + 2} \begin{bmatrix} \frac{\omega_{Mxy}}{\omega_F} y_1 \\ c\psi_1 \\ s\psi_1 \\ 0 \\ \frac{\omega_{Mxy}}{\omega_F} y_2 \\ c\psi_2 \\ s\psi_2 \\ 0 \end{bmatrix} \tag{17}$$

$$u_{M_y} = \frac{1}{\frac{\omega_{Mxy}}{\omega_F}\sum_{j=1}^{2} x_j^2 + 2} \begin{bmatrix} -\frac{\omega_{Mxy}}{\omega_F}x_1 \\ -s\psi_1 \\ c\psi_1 \\ 0 \\ -\frac{\omega_{Mxy}}{\omega_F}x_2 \\ -s\psi_2 \\ c\psi_2 \\ 0 \end{bmatrix}$$

$$u_{M_z} = \frac{1}{2}\left[0,0,0,1,0,0,0,1\right]^T$$

where $m_{pj} = m_p(1 - \left|\frac{x_j}{\bar{x_1}-\bar{x_2}}\right|)$, is the share of payload mass applied on each agent that rises when the asymmetric distribution of payload's CG is applied. $\frac{\omega_{Mxy}}{\omega_F}$ indicates the relationship between the moment and the force generated by the agents to produce a system moment. The bigger amount of $\frac{\omega_{Mxy}}{\omega_F}$ means the effect of agents forces is bigger than the effect of their moments in creating systems moment. x and y are the locations of the CG of each agent in body frame. Final control commands applying to each agent will be:

$$U = \left[u_F, u_{M_x}, u_{M_y}, u_{M_z}\right]\left[U_F, U_{M_x}, U_{M_y}, U_{M_z}\right]^T \quad (18)$$

This U is an 8×1 matrix which its first half elements are allocated to the first agent and the second half are allocated to the second agent.

4.2 Control Techniques

In order to control the attitude and altitude of the system we made use of two classic control laws which are PD and LQR methods. The control block of the system is described as:

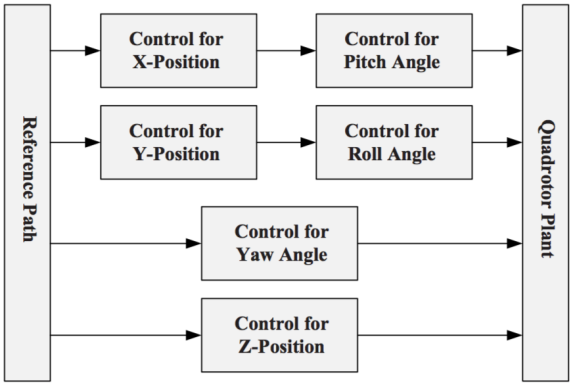


Figure 2: Control Logic

Reference path and controllers of X and Y positions will be described. Since the control laws are linear, the equations of motion of the system need to be linearized. Final linear equations are as follows :

$$\ddot{X} = \frac{-U_1\theta}{M}$$

$$\ddot{Y} = \frac{U_1\phi}{M} \quad (19)$$

$$\ddot{Z} = g - \frac{U_1}{M}$$

$$\ddot{\phi} = \frac{U_2}{I_X}$$

$$\ddot{\theta} = \frac{U_3}{I_Y} \quad (20)$$

$$\ddot{\psi} = \frac{U_4}{I_Z}$$

The control command inputs to each agent are as follows:

$$U_{j,F} = b(\Omega_{j,1}^2 + \Omega_{j,2}^2 + \Omega_{j,3}^2 + \Omega_{j,4}^2)$$

$$U_{j,M_x} = lb(\Omega_{j,4}^2 - \Omega_{j,2}^2) \quad (21)$$

$$U_{j,M_y} = lb(\Omega_{j,3}^2 - \Omega_{j,1}^2)$$

$$U_{j,M_z} = d(-\Omega_{j,1}^2 + \Omega_{j,2}^2 - \Omega_{j,3}^2 + \Omega_{j,4}^2)$$

4.2.1 PD Controller Design:

PD method is carried out for comparison with LQR method to see differences in performance of these two methods. The control command calculated by PD method is obtained as follows:

$$U = K_{(x)}(e_x) + K_{d(\dot{x})}(e_{(\dot{x})}) \quad (22)$$

Attitude control commands take the form of:

$$U_{2,3,4} = K_{p(\phi,\theta,\psi)}(e_{(\phi,\theta,\psi)}) + K_{d(\dot{\phi},\dot{\theta},\dot{\psi})}(e_{(\dot{\phi},\dot{\theta},\dot{\psi})}) \quad (23)$$

Altitude control output calculation is as explained above:

$$U_1 = K_{(z)}(e_z) + K_{d(\dot{z})}(e_{(\dot{z})}) \quad (24)$$

The PD gains are determined through Ziegler-Nichols method. The scheme of this controller is as shown in 'Fig. 3':

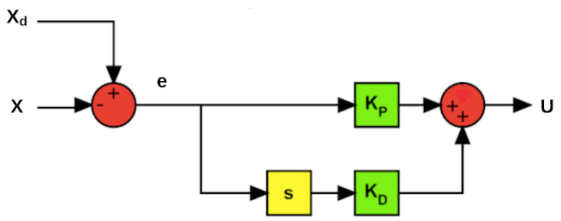


Figure 3: PD Controller Diagram

4.2.2 LQR Controller Design:

- LQR

Attitude Control:
First, we define the attitude state vectors as below:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix} \tag{25}$$

Then the attitude system can be written in state space form as:

$$\dot{X} = AX + BU \tag{26}$$

Which will be:

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{I_X} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{I_Y} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_Z} \end{bmatrix} \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{27}$$

The goal is to find the stabilizing feedback control law:

$$U = -KX \tag{28}$$

Which minimizes the cost function:

$$J = \int_0^\infty (X^T Q\, X + U^T R\, U) dt \tag{29}$$

where $Q \geq 0$ and $R > 0$ are weighting matrices of a appropriate dimensions. Q is related to the energy of the controlled output, while R is related to the energy of the control signal. So the choice of Q and R is a trade-off between the desired performance and the available capacities. Here we decided Q to be an identity matrix of appropriate dimension and R will be accomplished by a trial-and-error process until the **best** answer is obtained. Afterward, K is defined as:

$$K = R^{-1} B^T P \tag{30}$$

Where P is the solution of linear algebraic Riccati equation:

$$A^T P + PA - PBR^{-1} B^T P + Q = 0 \tag{31}$$

After calculating K controlling of the states will be accomplished.

Altitude Control:
For altitude the state vector is written as:

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} z \\ \dot{z} \end{bmatrix} \tag{32}$$

Then the state space form of the system is as:

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{M} \end{bmatrix} U_1 \tag{33}$$

All discussed about attitude control is applicable here too.

- Tracking LQR

Since the desired states of the system are not supposed to be always zero and the LQR control law is only applicable to the linear time-invariant systems, we needed to make use of $Tracking\ LQR$ controller. All discussed about LQR is applicable to $TLQR$ too but the control law will be different as follows [10]:

$$U = -K(X - X_d) \tag{34}$$

X_d is the desired state and can be either time-variant or time-invariant none-zero. Defining:

$$X_{d\ att} = \left[\phi_d, \dot{\phi_d}, \theta_d, \dot{\theta_d}, \psi_d, \dot{\psi_d}\right]^T \ ,\ X_{d\ alt} = \left[z_d, \dot{z_d}\right]^T$$

Calculation of desired rates will be discussed in 'Sec. 5'. For attitude and altitude control, after applying the iterative trial-and-error design procedure the following form for the Q and R matrices is reached:

Attitude:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{35}$$

$$R = \begin{bmatrix} 1 \times 10^{-3} & 0 & 0 \\ 0 & 1 \times 10^{-4} & 0 \\ 0 & 0 & 1 \times 10^{-3} \end{bmatrix} \tag{36}$$

Altitude:

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{37}$$

$$R = 6 \times 10^{-5} \tag{38}$$

5 Trajectory Generation

In addition to set point trajectory, a circular path was implemented and the stability and the performance of the system were analyzed. For both trajectories we made use of a PD controller which transforms the desired position to desired angle. This process is carried out by using Paparazzi

algorithm which consists of a reference generator and a PD controller and an attitude generator in order to convert the desired position to desired angle and desired angle to desired states which are the inputs of the control system. These algorithms are available at Paparazzi's website. In both paths its assumed that $\psi(t) = \psi(0) = 0$.

5.1 Paparazzi Set Point Reference Generator

The generator receives the desired position in X or Y direction as input signal and converts it to 3 output signals which are position, velocity and acceleration related to the desired point respectively. The scheme of the set point R.G. is shown below:

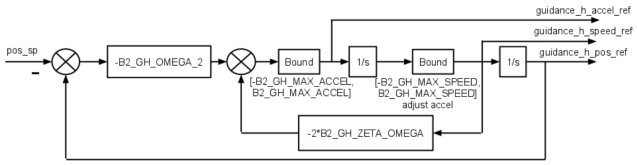


Figure 4: Set Point Reference Generator

5.2 Reference Controller and States Reference Generator

This module takes signals generated by 'Subsec. 5.1' as inputs and produces the desired attitude using a PD controller as follows:

$$(\theta_d) = K_p(X - X_{ref}) + K_d(\dot{X} - \dot{X}_{ref}) \qquad (39)$$

$$(\phi_d) = K_p(Y - Y_{ref}) + K_d(\dot{Y} - \dot{Y}_{ref}) \qquad (40)$$

As the control laws developed in 'Sec. 4' need desired states and considering that reference PD controller only generates the angle and not its rate, and noting that using the perfect derivation of this angle is not a scientific solution, we need to calculate the real attitude rate. This calculation is carried out by using Paparazzi attitude R.G. This module does the same as 'Subsec. 5.1' but the input is an angle not a position. This module is shown as:

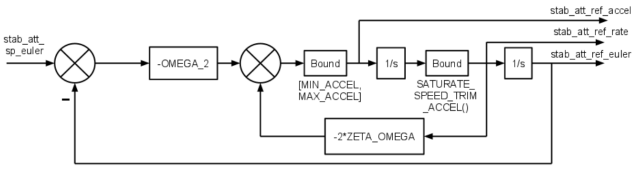


Figure 5: Attitude Reference Generator

It should be mentioned that 'Subsec. 5.1' is used in altitude controller too.

6 Results

In this section the behavior of the system's states in tracking two different trajectories will be analyzed and the performance of the controllers will be compared.

6.1 Set Point Trajectory

To make the simulation similar to what IMAV 2017 outdoor cooperative mission requires, the value of 50 meters is set as desired position. In order to see system's behavior in both lateral and longitudinal axes, the desired position is set in both X and Y directions and the results are as follows (fig.: 6-9):

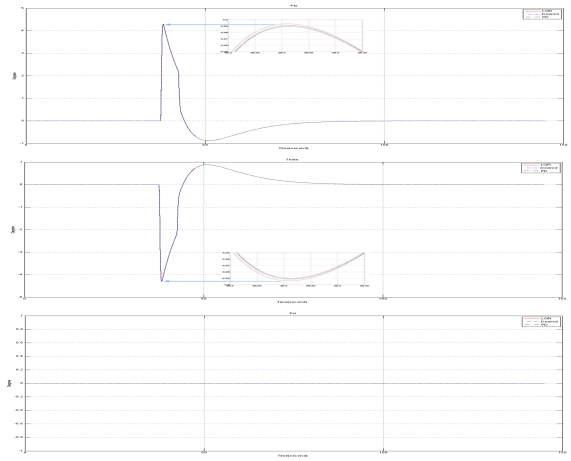


Figure 6: Euler Angles

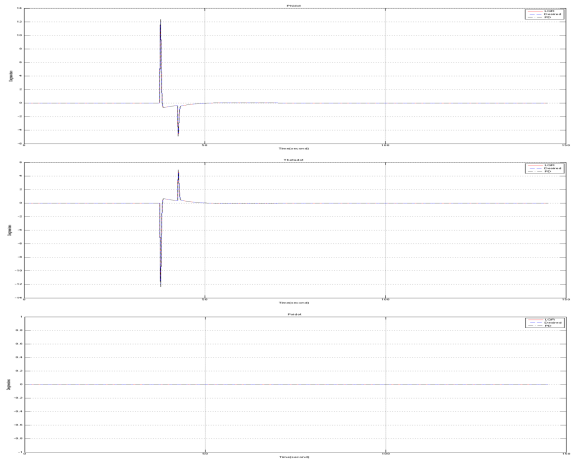


Figure 7: Euler Rates

6.2 Circular Trajectory

A circular path with a radius of 50 meters in an altitude of 3 meters is designed and is given to the system as the desired trajectory. The results are as follows (fig.: 10-13):

7 Conclusion and Future Works

Due to asymmetry in the geometry of the rigid-body system, for maneuvers in the asymmetry axis more control effort is needed to achieve the desired state and the stabilization in this axis is more challenging.

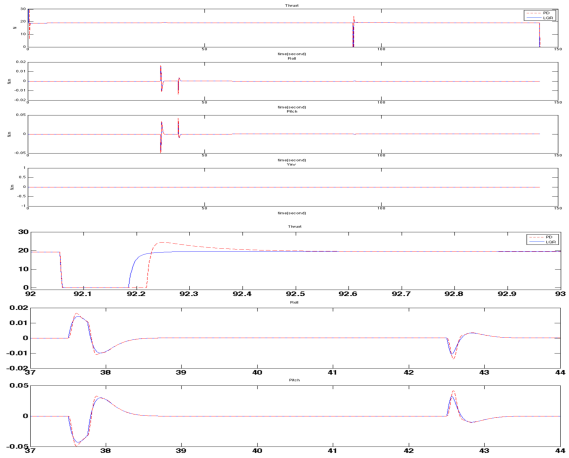


Figure 8: Control Inputs $(U_F, U_{M_{x,y,z}})$

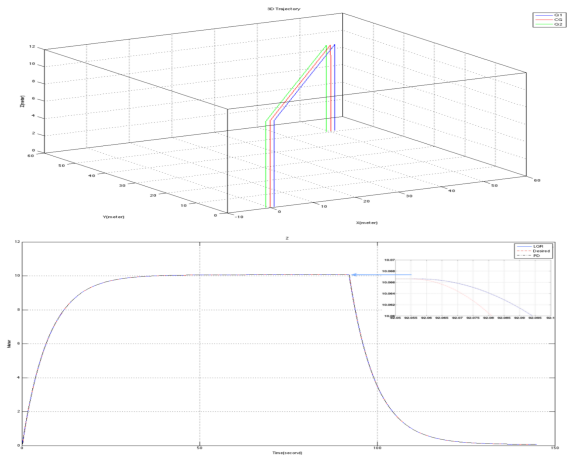


Figure 9: 3D Trajectory

We defined the subject of transporting a payload by 2 quadrotors rigidly attached to it and made use of two control techniques to reach the desired states of the system. Analyzing the achieved results in defined scenarios we can see that the behaviors of these two controllers are close to each other. We are currently planning to implement this simulation and carry out the experimental phase of this article using Paparazzi autopilot. Experimental results may be available in the final edition of this paper. Also a visualized simulation of this job is in progress.

Acknowledgements

The authors would like to thank Hamidreza Nemati for fruitful discussions and professional advises on $Traking\ LQR$ concept.

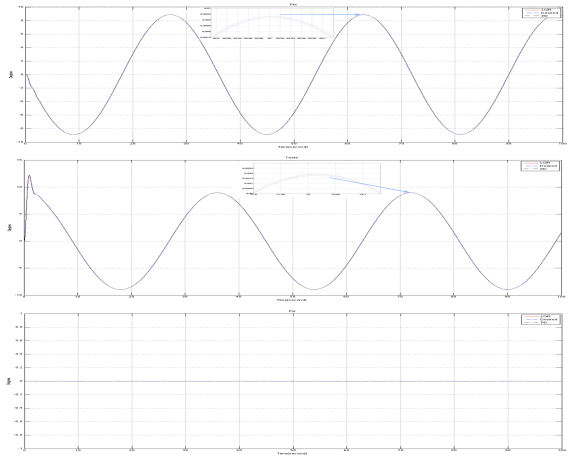


Figure 10: Euler Angles

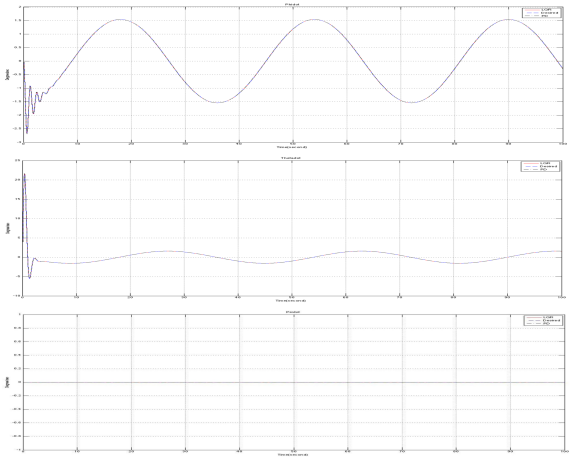


Figure 11: Euler Rates

References

[1] F.A. Goodarzi, D. Lee, and T. Lee. Geometric control of a quadrotor uav transporting a payload connected via flexible cable. *International Journal of Control, Automation and Systems*, 13(6):1486–1498, 2015.

[2] F.A. Goodarzii and T. Lee. Dynamics and control of quadrotor uavs transporting a rigid body connected via flexible cables. In *IEEE American Control Conference (ACC)*, 2015.

[3] N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, 2011.

[4] R. Oung and R. D'Andrea. The distributed flight array. *Mechatronics*, 21(6):908–917, 2011.

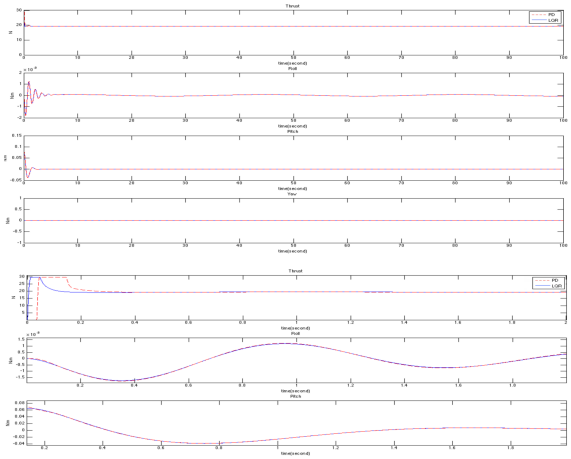


Figure 12: Control Inputs $(U_F, U_{M_{x,y,z}})$

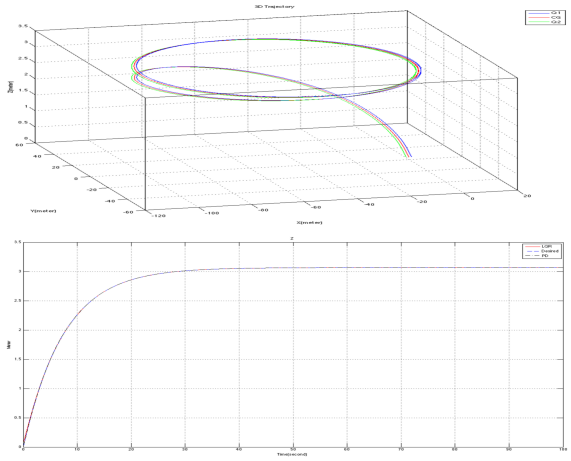


Figure 13: 3D Trajectory

[5] R. Ritz, M.W. Mller, M. Hehn, and R. DAndrea. Cooperative quadrocopter ball throwing and catching. In *IEEE Intelligent Robots and Systems (IROS)*, 2012.

[6] G. Gioioso, A. Franchi, G. Salvietti, S. Scheggi, and D. Prattichizzo. The flying hand: A formation of uavs for cooperative aerial tele-manipulation. In *IEEE Robotics and Automation (ICRA)*, 2014.

[7] V. Parra-Vega, A. Sanchez, C. Izaguirre, O. Garcia, and F. Ruiz-Sanchez. Toward aerial grasping and manipulation with multiple uavs. *Journal of Intelligent and Robotic Systems*, pages 1–19, 2013.

[8] A. Nikou, G.C. Gavridis, K.J. Kyriakopoulos, O. Garcia, , and F. Ruiz-Sanchez. Mechanical design, modelling and control of a novel aerial manipulator. In *IEEE Conference Robotics and Automation (ICRA)*, 2015.

[9] D. Mellinger, M. Shomin, N. Michael, and V. Kumar. *Cooperative grasping and transport using multiple quadrotors*. Springer Distributed autonomous robotic systems, Berlin, Heidelberg, 2013.

[10] V.G. Adir and A.M. Stocia. Integral lqr control of a star-shaped octorotor. *Incas Bulletin*, 4(2):3–18, 2012.

Robust Attitude Control for Quadrotors with External Disturbances

H. Nemati*, A. Naghash, S. Mozafari, and A. Jamei
Amirkabir University of Technology (Tehran Polytechnic), 424 Hafez Ave., Tehran 15875-4413, Iran

ABSTRACT

This study investigates a design procedure for a robust nonlinear control algorithm based on sliding mode control (SMC) to stabilize the attitude of a 3-DOF quadrotor UAV subject to external disturbances. Since traditional sliding mode controllers are sensitive against external disturbances in the reaching phase, a new algorithm is proposed to enhance the robust performance of an SMC strategy. Dynamic equations are obtained using Newton-Euler formalism and the quadrotor's centre of mass is assumed not to be coincident at the origin of body frame. The robust stability and the robust tracking property are achieved using the Lyapunov's direct method. Experimental results are given to highlight the effectiveness of the designed control strategy.

1 INTRODUCTION

Flight control of Unmanned Aerial Vehicles (UAVs) have received considerable attention from researchers recently because of numerous applications ranging from transportation (such as Amazon delivery system), security, rescue mission, agriculture, construction, traffic surveillance, image processing over nuclear reactors, management of natural risks, environment exploration, mapping, aerial cinematography, border prohibition and military. To accomplish these features which are executed in hazardous and inaccessible situations, the designed controller must be robust against environmental disturbances (such as wind gust) and parameter uncertainty (such as inertia variation).

Real systems exhibit hard nonlinearities such as Coulomb friction, actuator saturation, valve dead zones, gear backlash and hysteresis which may possess some discontinuous features that do not lend themselves to the linear approximation [1]. Nonlinear control systems provide a level of dynamic capabilities when dealing with parameter variations and unmodeled dynamics that linear approaches cannot deal with. Indeed, the success of a UAV mission strongly depends upon the precision of its attitude control in spite of the presence of environmental disturbances and large parameter uncertainties. Hence, the designed controller must be somewhat immune to variations across parameters in the model and be able to overcome the above inaccuracies. Sliding mode control (SMC) has been recognized as a robust control technique for quadrotor's attitude motion because of its inherent advantages of strong stability, disturbance rejection and low sensitivity to plant parameter variations [2].

The quadrotor is a four-propeller Vertical Take-Off and Landing (VTOL) rotorcraft which has been proven to be one of the efficient vehicles to achieve rapid turns and strong maneuverability compared to traditional aircraft. Over several years much research has been devoted to design of the attitude controller for the quadrotor UAV. In [3], a robust flight controller for a 6-DOF quadrotor model has been designed based on the sliding mode control driven by sliding mode disturbance observer (SMC-SMDO). This controller has been relied on the knowledge of the limits of the disturbance. Since the determination of the coefficients used in the sliding surface is difficult in practice, Hurwitz stability analysis has been employed in [4, 5] to obtain the nonlinear coefficients of the second order sliding manifold. Besides, the nonlinear sliding surface has been simplified by linearizing around the desired equilibrium points and then the nonlinear coefficients were calculated by Hurwitz stability. A super twisting sliding mode controller has been designed in [6] by utilizing a cascaded inner-outer loop structure for a quadrotor. Its robustness has been also compared against a traditional SMC, a popular linear controller (LQR-PD) and a nonlinear feedback linearization based controller subject to wind turbulence conditions and modeling uncertainties. Active disturbance rejection control are often used to eliminate the effect of the state coupling and uncertainties. The robust trajectory tracking problem of an autonomous quadrotor with obstacle avoidance based on the active disturbance rejection control has been introduced in [7]. The problem of attitude regulation for a quadrotor with parametric uncertainties and external disturbances has been studied theoretically by employing a novel adaptive fuzzy gain-scheduling sliding mode control approach in [8].

So far, a large number of references have been devoted to the theoretical analysis of a quadrotor flight controller based upon the linear and nonlinear control methods. There is much research for position/attitude control of a quadrotor in a real time. Model based controller for position and attitude trajectory tracking of a quadrotor have been introduced experimentally in [9]. The problem of designing and experimentally validating a controller based on a backstepping procedure for steering a quadrotor system along a trajectory subject to ex-

*Email address: hr.nemati@gmail.com

ternal disturbances has been addressed in [10, 11]. A high-performance flight control approach utilizing an active disturbance rejection technique for the quadrotors has been studied in [12]. Recently, second order sliding mode controllers which are the simplest class of the higer order sliding mode (HOSM) approach are taken into account experimentally by many researchers. A popular type of the second order sliding mode control, super twisting algorithm, has been utilized in [13] to stabilize a quadrotor UAV experimentally. However, HOSM is not able to ensure the finite time stability [14] and the system trajectory in the second order sliding mode control strategy is very slow when states are far away from the origin [15]. Traditional SMC has been employed in [16] to stabilize the translational motion of a quadrotor experimentally.

Generally, sliding motion consists of two phases: reaching phase and sliding phase. In the reaching phase, the system trajectory starts from a given initial condition of the predetermined sliding surface, moves towards it and reaches it in a finite time. In the sliding phase, the motion is restricted to the sliding manifold, lies on it and converges to the desired condition. However, the control system trajectory in the reaching phase is sensitive to external disturbances and uncertainties while the system motion is insensitive against disturbances/parameter variations within sliding phase. There exists a method to enhance the robust performance of the SMC technique by shortening the reaching phase known as Time-Varying Sliding Mode Control (TVSMC). In [17] a stepwise time-varying switching manifold has been introduced. However, this method cannot guarantee insensitivity of the system subject to external disturbances and parameter uncertainties.

In this paper, the problem of attitude control of a 3-DOF quadrotor UAV is investigated experimentally. Accurate quadrotor model is considered such that the centre of mass of the quadrotor does not coincide with the origin of the body-fixed frame. The reminder of this paper is organized as follows: at first, preliminaries for deriving an attitude model of a 3-DOF quadrotor are introduced. Then, a design procedure of an SMC law is developed. Thereafter, numerical simulations and experimental implementations are performed to validate the effectiveness of the designed controller in the presence of the wind gust (as an external disturbance) using an experimental wind tunnel. Finally, this paper is ended with some concluding remarks.

2 Dynamic Model

The dynamical model of a quadrotor consisting of four propellers in cross-shaped frame is studied in this section, as shown in Fig. 1. The attitude change of the quadrotor results from variations on forces and moments produced by adjusting rotors' speeds. To calculate the dynamic model of the quadrotor, the following assumptions have been considered:

- The quadrotor's structure is supposed to be rigid and symmetrical.

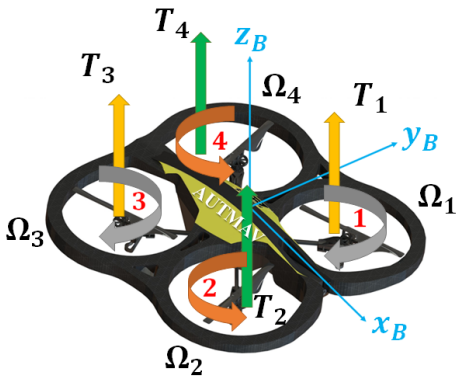


Figure 1: Schematic of the quadrotor configuration with coordinate axes

- The propellers are rigid *i.e.* propeller flapping does not happen.
- The centre of mass does not coincide with the origin of the body-fixed frame.
- Aerodynamic forces and moments are proportional to the square of rotor's speed.
- The axes of the body-frame are coincident with the principle axes of the quadrotor *i.e.* the inertia matrix of the quadrotor is diagonal.

Let $\mathscr{E} = \{x_E, y_E, z_E\}$ be the Earth-fixed inertial frame and $\mathscr{B} = \{x_B, y_B, z_B\}$ denotes the body-fixed frame in which its origin is located in the centre of mass of the quadrotor. The attitude of the quadrotor is described by ZYX Euler angle notations where the Euler angles $\Theta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ are respectively known as roll (rotation around $x-$axis), pitch (rotation around $y-$axis) and yaw (rotation around $z-$axis). Attitude angles are bounded as follows: $\phi \in (-\frac{\pi}{2}, \frac{\pi}{2})$, $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ and $\psi \in (-\pi, \pi)$ because the various acrobatic flying is not admissible. $\omega^B = \begin{bmatrix} p & q & r \end{bmatrix}^T$ represents quadrotor's angular velocity in frame $\mathscr{B}$. The rotational kinematics is obtained from the transformation of the Euler rates $\dot{\Theta} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$ measured in the Earth-fixed inertial frame and angular body rates $\omega^B = \begin{bmatrix} p & q & r \end{bmatrix}^T$ as follows:

$$\omega^B = \mathcal{M}\,\dot{\Theta}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & S_\phi\, C_\theta \\ 0 & -S_\phi & C_\phi\, C_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (1)$$

where "S" and "C" denote "sin" and "cos" trigonometric functions, respectively. Euler angles are assumed to be small around the hovering position. On the other hand, $\cos\phi = \cos\theta = 1$, $\sin\phi = \phi$ and $\sin\theta = \theta$ are assumed.

The attitude dynamics of the quadrotor can be derived using the Newton-Euler equations in the body frame with the following general formalism:

$$J\dot{\omega}^B + \omega^B \times (J\omega^B) + M_g = \boldsymbol{\tau} \tag{2}$$

where J is an inertia matrix $\Big(J = \text{diag}[J_{xx}, J_{yy}, J_{zz}]\Big)$ of the quadrotor, M_g is the propeller gyroscopic effect and $\boldsymbol{\tau} = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T$ presents the total moments acting on the quadrotor in the body frame. The gyroscopic moment resulting from the propeller's kinetic energies can be described as:

$$M_g = \omega^B \times \begin{bmatrix} 0 & 0 & J_r\Omega_r \end{bmatrix}^T \tag{3}$$

where J_r denotes the propellers' inertia and Ω_r represents the relative propeller's speed which can be defined as:

$$\Omega_r = \sum_{i=1}^{4} (-1)^{i+1}\Omega_i \tag{4}$$

The aerodynamic forces (T) and moments (M) produced by the ith propeller can be expressed as:

$$\begin{aligned} T_i &= k_{Ti}\,{\Omega_i}^2 \\ M_i &= k_{Mi}\,{\Omega_i}^2 \end{aligned} \tag{5}$$

where k_{Ti} and k_{Mi} are aerodynamic constants which can be determined experimentally for each rotor. Then, the total moments $\boldsymbol{\tau}$ can be derived using the following relationship:

$$\begin{cases} \tau_x = \dfrac{\sqrt{2}}{2} L\,(T_1 - T_2 - T_3 + T_4) \\ \tau_y = \dfrac{\sqrt{2}}{2} L\,(-T_1 - T_2 + T_3 + T_4) \\ \tau_z = M_1 - M_2 + M_3 - M_4 \end{cases} \tag{6}$$

where L is a distance between the propeller and the centre of mass of the quadrotor. Hence, control inputs can be described by combining Eqs. (5) and (6) in a vector form as:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{\sqrt{2}}{2}L & -\frac{\sqrt{2}}{2}L & -\frac{\sqrt{2}}{2}L & \frac{\sqrt{2}}{2}L \\ -\frac{\sqrt{2}}{2}L & -\frac{\sqrt{2}}{2}L & \frac{\sqrt{2}}{2}L & \frac{\sqrt{2}}{2}L \\ \frac{k_{M1}}{k_{T1}} & -\frac{k_{M2}}{k_{T2}} & \frac{k_{M3}}{k_{T3}} & -\frac{k_{M4}}{k_{T4}} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \tag{7}$$

Eventually, the quadrotor attitude dynamics can be written using Eqs. (2) and (7) in the following form:

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{J_{yy}+mc^2-J_{zz}}{J_{xx}+mc^2}\dot{\theta}\dot{\psi} \\ \frac{J_{zz}-J_{xx}-mc^2}{J_{yy}+mc^2}\dot{\phi}\dot{\psi} \\ \frac{J_{xx}-J_{yy}}{J_{zz}}\dot{\phi}\dot{\theta} \end{bmatrix} - J_r\Omega_r \begin{bmatrix} \frac{\dot{\theta}}{J_{xx}+mc^2} \\ -\frac{\dot{\phi}}{J_{yy}+mc^2} \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{u_2}{J_{xx}+mc^2} \\ \frac{u_3}{J_{yy}+mc^2} \\ \frac{u_4}{J_{zz}} \end{bmatrix} + mgc \begin{bmatrix} -\frac{\sin\phi}{J_{xx}+mc^2} \\ -\frac{\sin\theta}{J_{yy}+mc^2} \\ 0 \end{bmatrix} \tag{8}$$

where c is a distance between the centre of mass and the centre of rotation of the quadrotor. Since propellers are very light, their moment of inertia can be ignored.

3 Controller Design

In this section, sliding mode control (SMC) is designed for stabilization of the attitude of an over-actuated 3-DOF quadrotor.

3.1 Conventional Sliding Mode Control (Conv. SMC)

A conventional sliding surface ($\sigma_{conv.}$) can be expressed as follows:

$$\sigma_{conv.,i} = \dot{\tilde{x}}_i + \lambda_i\,\tilde{x}_i \qquad ; \qquad i = 1, 2, 3 \tag{9}$$

where $x_i = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ is a state vector, $\tilde{x}$ is a perturbation from the reference ($\tilde{x} = x - x_d$) and λ is a positive constant.

To guarantee the asymptotic stability of the system, consider the following positive-definite function of $\sigma_{conv.}$ as a Lyapunov candidate:

$$V_i = \frac{1}{2}\,{\sigma_{conv.,i}}^2 \tag{10}$$

The time derivative of the Lyapunov candidate can be written in the following form:

$$\begin{aligned} \dot{V}_i &= \sigma_{conv.,i}\,\frac{\partial \sigma_{conv.,i}}{\partial t} \\ &= \sigma_{conv.,i}\,(\ddot{\tilde{x}}_i + \lambda_i\,\dot{\tilde{x}}_i) \end{aligned} \tag{11}$$

According to the Lyapunov's direct method, the negative definiteness of $\dot{V}$ implies that the equilibrium state at the origin is asymptotically stable *i.e.*,

$$\begin{aligned} &\sigma_{conv.,i}\,\frac{\partial \sigma_{conv.,i}}{\partial t} = -\mu_i\,|\sigma_{conv.,i}| \\ \Rightarrow\ &\ddot{\tilde{x}}_i + \lambda_i\,\dot{\tilde{x}}_i = -\mu_i\ \text{sign}\,\sigma_{conv.,i} \end{aligned} \tag{12}$$

where μ is a positive constant. $\text{sign}\,\sigma$ denotes a signum function and can be determined as:

$$\text{sign}\,\sigma = \begin{cases} +1 & \sigma > 0 \\ 0 & \sigma = 0 \\ -1 & \sigma < 0 \end{cases} \tag{13}$$

As a consequence, the nonlinear control inputs are designed using Eqs. (8) and (12) as:

$$\begin{aligned} u_2 =& (J_{xx} + mc^2)\,\Big[\ddot{\phi}_d - \mu_1\,\text{sign}\,\sigma_{conv.,1} - \lambda_1(\dot{\phi} - \dot{\phi}_d)\Big] + \\ & (J_{zz} - J_{yy} - mc^2)\,\dot{\theta}\dot{\psi} + J_r\Omega_r\dot{\theta} + mgc\,\sin\phi \end{aligned} \tag{14}$$

,

$$\begin{aligned} u_3 =& (J_{yy} + mc^2)\,\Big[\ddot{\theta}_d - \mu_2\,\text{sign}\,\sigma_{conv.,2} - \lambda_2(\dot{\theta} - \dot{\theta}_d)\Big] + \\ & (J_{xx} + mc^2 - J_{zz})\,\dot{\phi}\dot{\psi} - J_r\Omega_r\dot{\phi} + mgc\,\sin\theta \end{aligned} \tag{15}$$

and

$$u_4 = J_{zz}\left[\ddot{\psi}_d - \mu_3 \operatorname{sign}\sigma_{conv.,3} - \lambda_3(\dot{\psi} - \dot{\psi}_d)\right] + (J_{yy} - J_{xx})\,\dot{\phi}\dot{\theta} \tag{16}$$

As it can be seen from Eqs. (13) to (16), signum function is a discontinuous function which can switch the control signal at an infinite frequency and therefore excite the unmodeled fast dynamics or undesired oscillations called *chattering* (or *ripple*). The chattering phenomenon is undesired in practice because it can reduce the system's performance or even lead to instability. Particularly, for a system with flexible appendages, the chattering may disintegrate the whole system. To solve this problem, one may approximate the discontinuous function with a hyperbolic switching function as:

$$u_2 = (J_{xx} + mc^2)\left[\ddot{\phi}_d - \mu_1 \tanh\sigma_{conv.,1} - \lambda_1(\dot{\phi} - \dot{\phi}_d)\right] + (J_{zz} - J_{yy} - mc^2)\,\dot{\theta}\dot{\psi} + J_r\Omega_r\dot{\theta} + mgc\,\sin\phi \tag{17}$$

,

$$u_3 = (J_{yy} + mc^2)\left[\ddot{\theta}_d - \mu_2 \tanh\sigma_{conv.,2} - \lambda_2(\dot{\theta} - \dot{\theta}_d)\right] + (J_{xx} + mc^2 - J_{zz})\,\dot{\phi}\dot{\psi} - J_r\Omega_r\dot{\phi} + mgc\,\sin\theta \tag{18}$$

and

$$u_4 = J_{zz}\left[\ddot{\psi}_d - \mu_3 \tanh\sigma_{conv.,3} - \lambda_3(\dot{\psi} - \dot{\psi}_d)\right] + (J_{yy} - J_{xx})\,\dot{\phi}\dot{\theta} \tag{19}$$

3.2 Nonsingular Terminal Sliding Mode Control (NTSMC)

Conventional SMC utilizes a linear switching surface which cannot guarantee the finite time convergence of the state variables. On the other hand, the convergence rate to the desired state variables has an infinite settling time exponentially. In [18], a hierarchical control strategy has been presented theoretically based on the adaptive radical basis function neural networks and an integral SMC for the position and attitude tracking of a quadrotor UAV. Though, the hierarchical control scheme offered fast convergence, it employs larger control domain. To enhance the convergence performance of the traditional sliding mode controller, terminal sliding mode control (TSMC) has been introduced [19]. TSMC uses nonlinear switching manifold in which output errors converge to zero in a finite time. However, TSMC has a critical disadvantage of a singularity problem. Hence, nonsingular terminal sliding mode control (NTSMC) [20] is employed to stabilize the attitude of a quadrotor UAV.

Without loss of generality, a nonsingular terminal sliding manifold can be described as:

$$\sigma_{NTSMC,i} = \tilde{x}_i + \xi_i\,\dot{\tilde{x}}_i^{\frac{\beta_i}{\alpha_i}} \quad ; \quad i = 1,2,3 \tag{20}$$

where α and β are positive odd integers and must satisfy $\alpha < \beta < 2\alpha$. ξ is a positive scalar. The nonlinearity term $\dot{\tilde{x}}_i^{\frac{\beta}{\alpha}}$ in Eq. (20) can improve the convergence speed and assure a bounded control input toward an equilibrium point. To guarantee that the state trajectory remains around the nonsingular terminal switching surface, the following condition should be satisfied:

$$\sigma_{NTSMC,i}\,\frac{\partial\sigma_{NTSMC,i}}{\partial t} < 0 \tag{21}$$

Taking the time-derivative of Eq. (21) along Eq. (20) leads to:

$$\sigma_{NTSMC,i}\,\frac{\partial\sigma_{NTSMC,i}}{\partial t} = \sigma_{NTSMC,i}\left(\dot{\tilde{x}}_i + \xi_i\frac{\beta_i}{\alpha_i}\,\ddot{\tilde{x}}_i\dot{\tilde{x}}_i^{\,(\frac{\beta_i}{\alpha_i}-1)}\right) \tag{22}$$

As a result, to ensure the Lyapunov stability, the following relation should be satisfied:

$$\dot{\tilde{x}}_i^{\,(2-\frac{\beta_i}{\alpha_i})} + \xi_i\frac{\beta_i}{\alpha_i}\,\ddot{\tilde{x}}_i = -\mu_i\,\operatorname{sign}\sigma_{NTSMC,i} \tag{23}$$

Finally, nonsingular terminal sliding mode controllers can be rewritten as:

$$u_2 = \left(\ddot{\phi}_d - \mu_1 \operatorname{sign}\sigma_{NTSMC,1} - (\dot{\phi} - \dot{\phi}_d)^{(2-\frac{\beta_1}{\alpha_1})}\right)(J_{xx} + mc^2)\frac{\alpha_1}{\xi_1\beta_1} + (J_{zz} - J_{yy} - mc^2)\,\dot{\theta}\dot{\psi} + J_r\Omega_r\dot{\theta} + mgc\,\sin\phi \tag{24}$$

,

$$u_3 = \left(\ddot{\theta}_d - \mu_2 \operatorname{sign}\sigma_{NTSMC,2} - (\dot{\theta} - \dot{\theta}_d)^{(2-\frac{\beta_2}{\alpha_2})}\right)(J_{yy} + mc^2)\frac{\alpha_2}{\xi_2\beta_2} + (J_{xx} + mc^2 - J_{zz})\,\dot{\phi}\dot{\psi} - J_r\Omega_r\dot{\phi} + mgc\,\sin\theta \tag{25}$$

and

$$u_4 = \left(\ddot{\psi}_d - \mu_3 \operatorname{sign}\sigma_{NTSMC,3} - (\dot{\psi} - \dot{\psi}_d)^{(2-\frac{\beta_3}{\alpha_3})}\right)J_{zz}\frac{\alpha_3}{\xi_3\beta_3} + (J_{yy} - J_{xx})\,\dot{\phi}\dot{\theta} \tag{26}$$

As can be observed from Eqs. (24) to (26), nonsingular terminal sliding controllers still include the discontinuous signum function which may cause the chattering phenomenon. To avoid this problem, the signum function can be easily replaced by the hyperbolic function as discussed from Eqs. (14) to (16).

$$u_2 = \left(\ddot{\phi}_d - \mu_1 \tanh\sigma_{NTSMC,1} - (\dot{\phi} - \dot{\phi}_d)^{(2-\frac{\beta_1}{\alpha_1})}\right)(J_{xx} + mc^2)\frac{\alpha_1}{\xi_1\beta_1} + (J_{zz} - J_{yy} - mc^2)\,\dot{\theta}\dot{\psi} + J_r\Omega_r\dot{\theta} + mgc\,\sin\phi \tag{27}$$

,

$$u_3 = \Big(\ddot{\theta}_d - \mu_2 \tanh \sigma_{NTSMC,2} - (\dot{\theta} - \dot{\theta}_d)^{(2-\frac{\beta_2}{\alpha_2})}\Big) (J_{yy} + mc^2)\frac{\alpha_2}{\xi_2\beta_2} + (J_{xx} + mc^2 - J_{zz})\,\dot{\phi}\dot{\psi} - J_r\Omega_r\dot{\phi} + mgc\,\sin\theta \quad (28)$$

and

$$u_4 = \Big(\ddot{\psi}_d - \mu_3 \tanh \sigma_{NTSMC,3} - (\dot{\psi} - \dot{\psi}_d)^{(2-\frac{\beta_3}{\alpha_3})}\Big) J_{zz}\frac{\alpha_3}{\xi_3\beta_3} + (J_{yy} - J_{xx})\,\dot{\phi}\dot{\theta} \quad (29)$$

3.3 Slope-Varying Nonsingular Terminal Sliding Mode Control (SVNTSMC)

In this subsection, a new continuous time-varying switching surface is introduced. The main feature of the proposed time-varying sliding mode control approach is to shorten the reaching phase via continuously rotating switching manifold (known as slope-varying). Therefore, the following sliding manifold is suggested as:

$$\sigma_{SVNTSMC,i} = \tilde{x}_i + f_i(t)\,\xi_i\,\dot{\tilde{x}}_i^{\frac{\beta_i}{\alpha_i}} \quad ; \quad i = 1,2,3 \quad (30)$$

where $f_i(t)$ is a nonlinear time-varying function. It should be noted that the proposed time-varying switching manifold is applicable to any types of SMC approaches, although NTSMC has been chosen because of its aforementioned advantages.

Then, the design procedure of the proposed nonlinear function $f(t)$ is studied. First, the initial values for $f(t)$ function must be described such that the initial states lies on the sliding surface i.e.,

$$f_i(0) = -\frac{\tilde{x}_i(0)}{\xi_i\,\dot{\tilde{x}}_i(0)^{\frac{\beta_i}{\alpha_i}}} \quad ; \quad i = 1,2,3 \quad (31)$$

Thereafter, the function $f(t)$ must be chosen such that the slope-varying sliding manifold approaches the desired sliding surface in a finite time. As an example, one may select the following nonlinear function:

$$f_i(t) = f_i(0) + \Big(1 - f_i(0)\Big)\tanh t \quad ; \quad i = 1,2,3 \quad (32)$$

As a result, slope-varying nonsingular terminal sliding mode controllers can be derived as:

$$u_2 = \Big(\ddot{\phi}_d - \mu_1 \tanh \sigma_{SVNTSMC,1} - (\dot{\phi} - \dot{\phi}_d)^{(2-\frac{\beta_1}{\alpha_1})} - \xi_1\,\dot{f}_1(t)(\dot{\phi} - \dot{\phi}_d)\Big)(J_{xx} + mc^2)\frac{\alpha_1}{\xi_1\beta_1\,f_1(t)} + (J_{zz} - J_{yy} - mc^2)\,\dot{\theta}\dot{\psi} + J_r\Omega_r\dot{\theta} + mgc\,\sin\phi \quad (33)$$

,

$$u_3 = \Big(\ddot{\theta}_d - \mu_2 \tanh \sigma_{SVNTSMC,2} - (\dot{\theta} - \dot{\theta}_d)^{(2-\frac{\beta_2}{\alpha_2})} - \xi_2\,\dot{f}_2(t)(\dot{\theta} - \dot{\theta}_d)\Big)(J_{yy} + mc^2)\frac{\alpha_2}{\xi_2\beta_2\,f_2(t)} + (J_{xx} + mc^2 - J_{zz})\,\dot{\phi}\dot{\psi} - J_r\Omega_r\dot{\phi} + mgc\,\sin\theta \quad (34)$$

and

$$u_4 = \Big(\ddot{\psi}_d - \mu_3 \tanh \sigma_{SVNTSMC,3} - (\dot{\psi} - \dot{\psi}_d)^{(2-\frac{\beta_3}{\alpha_3})} - \xi_3\,\dot{f}_3(t)(\dot{\psi} - \dot{\psi}_d)\Big) J_{zz}\frac{\alpha_3}{\xi_3\beta_3\,f_3(t)} + (J_{yy} - J_{xx})\,\dot{\phi}\dot{\theta} \quad (35)$$

To calculate a first component of the control input vector (u_1), let's rewrite T_i from Eq. (7) as:

$$\begin{aligned} T_1 &= \frac{1}{4}u_1 + \frac{\sqrt{2}}{4L}u_2 - \frac{\sqrt{2}}{4L}u_3 + \frac{k_{T1}}{4\,k_{M1}}u_4 \\ T_2 &= \frac{1}{4}u_1 - \frac{\sqrt{2}}{4L}u_2 - \frac{\sqrt{2}}{4L}u_3 - \frac{k_{T2}}{4\,k_{M2}}u_4 \\ T_3 &= \frac{1}{4}u_1 - \frac{\sqrt{2}}{4L}u_2 + \frac{\sqrt{2}}{4L}u_3 + \frac{k_{T3}}{4\,k_{M3}}u_4 \\ T_4 &= \frac{1}{4}u_1 + \frac{\sqrt{2}}{4L}u_2 + \frac{\sqrt{2}}{4L}u_3 - \frac{k_{T4}}{4\,k_{M4}}u_4 \end{aligned} \quad (36)$$

It is apparent from Eq. (36) that each T_i includes a fixed term ($\frac{1}{4}u_1$) and a variable term ($\bar{u}_i$) as:

$$\begin{aligned} \bar{u}_1 &= \frac{\sqrt{2}}{4L}u_2 - \frac{\sqrt{2}}{4L}u_3 + \frac{k_{T1}}{4\,k_{M1}}u_4 \\ \bar{u}_2 &= -\frac{\sqrt{2}}{4L}u_2 - \frac{\sqrt{2}}{4L}u_3 - \frac{k_{T2}}{4\,k_{M2}}u_4 \\ \bar{u}_3 &= -\frac{\sqrt{2}}{4L}u_2 + \frac{\sqrt{2}}{4L}u_3 + \frac{k_{T3}}{4\,k_{M3}}u_4 \\ \bar{u}_4 &= \frac{\sqrt{2}}{4L}u_2 + \frac{\sqrt{2}}{4L}u_3 - \frac{k_{T4}}{4\,k_{M4}}u_4 \end{aligned} \quad (37)$$

Therefore, the minimum value of u_1 can be obtained as:

$$u_1 = 4\,\Big|(\min\,\bar{u}_i) - T_{min}\Big| \quad (38)$$

where T_{min} is a minimum thrust that can be computed experimentally as $T_{min} = 0.096$ N.

4 Simulation and Experimental Results

This section is dedicated to simulation and real-time experimental validation of the nonsingular terminal sliding mode control (NTSMC) approach for stabilizing the attitude of a 3-DOF quadrotor UAV. The designed experimental test

bed is demonstrated in Fig. 2. The quadrotor under investigation consists of an inertial measurement unit (IMU) and an internal computer with a 1 GHz 32 bit ARM Cortex A8 processor. All the measurements collected by the micro-processor with a frequency of 500 Hz is sent to a ground station using a Wi-Fi connection. The testing architecture is developed utilizing a Paparazzi UAV to merge sensors, control law and the communication with the quadrotor. The identified parameters for the experimental test bed are as follows: $m = 0.42$ kg, $L = 0.18$ m, $J_{xx} = J_{yy} = 1.8 \times 10^{-3}$ kg m^2, $J_{zz} = 4.7 \times 10^{-3}$ kg m^2, $J_r = 1.85 \times 10^{-5}$ kg m^2, $k_{Ti} = 5.7 \times 10^{-6}$, $k_{Mi} = 1.7 \times 10^{-7}$, $g = 9.81$ m/s^2 and $c = 3 \times 10^{-2}$ m. In this work, attitude Euler angles are restricted as: $-\frac{\pi}{3} \leq \phi \leq \frac{5\pi}{12}$ and $-\frac{\pi}{3} \leq \theta \leq \frac{\pi}{3}$. Sliding gains are chosen as $\xi_1 = \xi_2 = \xi_3 = 0.5$, $\mu_1 = \mu_2 = \mu_3 = 6$, $\alpha_1 = \alpha_2 = \alpha_3 = 11$ and $\beta_1 = \beta_2 = \beta_3 = 13$.

In this paper, two sliding mode controllers (NTSMC and the proposed SVNTSMC) are applied to a quadrotor subject to the maximum wind velocity (18 m/s). Then, a roll command ($\phi_d = 10$ deg) is executed to the system and attitude stabilization is performed during this maneuver. It is necessary to mention that the wind gust is provided in the AUT wind tunnel, DANA Laboratory.

Attitude tracking of the quadrotor by experiments are shown in Figs. 3 and 4. Fig. 3 displays time history of Euler angles experimentally using NTSMC. Time history of Euler rates are demonstrated in Figs. 5 and 6 based on NTSMC and the proposed SVNTSMC approaches, respectively. From Figs. 3 and 5, one can simply observe that the NTSMC approach is not able to handle large disturbance at the final time. Figs. 4 and 6 prove that the proposed SVNTSMC could successfully enhance the robust tracking performance. In addition, there are some high frequencies present in practice such as unmodeled dynamics, sensor noises and so on that show up in the experimental results.



Figure 2: Experimental test bed of a 3-DOF quadrotor in the AUTMAV laboratory.

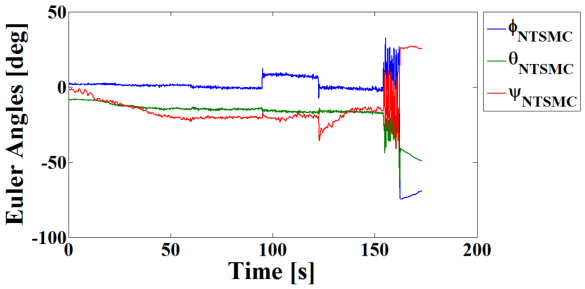


Figure 3: Time history of Euler angles using NTSMC.

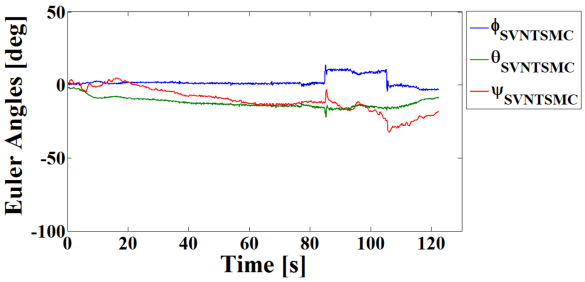


Figure 4: Time history of Euler angles using the proposed SVNTSMC.

5 Conclusion

In this paper, an effective implementation of a nonsingular terminal sliding mode control (NTSMC) approach has been developed for attitude stabilization of a 2-DOF quadrotor. Furthermore, the mathematical model of a quadrotor is derived accurately by considering that the centre of mass of the quadrotor does not coincide with the origin of the body-fixed frame. Traditional sliding mode controllers are not able to tackle any disturbances in the reaching phase. Therefore, a new sliding mode control approach known as Slope-Varying Nonsingular Terminal Sliding Mode Control (SVNTSMC) is proposed to enhance the robust performance of the plant subject to external disturbances. The applied NTSMC and SVNTSMC can assure the finite convergence time of the

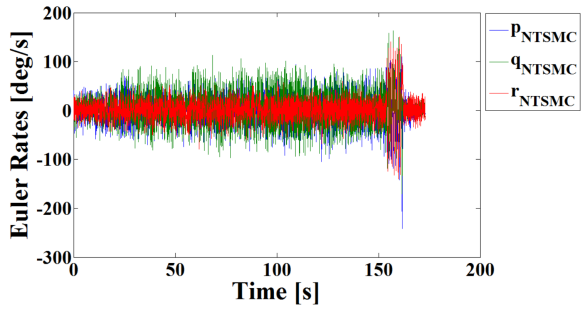


Figure 5: Time history of Euler rates using NTSMC.

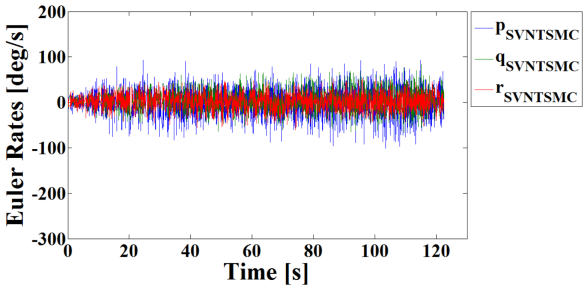


Figure 6: Time history of Euler rates using the proposed SVNTSMC.

state variables as well as singularity avoidance. The chattering phenomenon is eliminated using a smooth approximation of the switching surface around origin. Finally, NTSMC and the proposed SVNTSMC are applied experimentally to a quadrotor subject to a wind gust (as an external disturbance) with high velocity. The comparison between two controllers proved that the proposed SVNTSMC law is effective in practice.

Acknowledgements

This study is supported by National Elites Foundation (known as Bonyad Melli Nokhbegan, BMN, in Persian), Vice-Presidency for Science and Technology, Iran (under grant: 20.152). Authors would like to thank BMN for its support. In addition, gratitude is given to Prof. Mahmoud Mani (Head of Experimental Aerodynamics Research Laboratory known as DANA, Department of Aerospace Engineering, Amirkabir Uniersity of Technology) who graciously allowed us to utilize the wind tunnel setup in the DANA laboratory.

References

[1] J.-J. E. Slotine and W.P. Li. *Applied nonlinear control*. Prentice Hall, Englewood cliffs, 1991.

[2] V.I. Utkin. *Sliding modes in control and optimization*. Springer, New York, NY, 1992.

[3] L. Besnard, Y.B. Shtessel, and B. Landrum. Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer. *Journal of the Franklin Institute*, 349(2):658–684, 2012.

[4] E.-H. Zheng, J.-J. Xiong, and J.-L. Juo. Second order sliding mode control for a quadrotor uav. *ISA Transactions*, 53(4):1350–1356, 2014.

[5] J.-J. Xiong and G.-B. Zhang. Global fast dynamic terminal sliding mode control for a quadrotor uav. *ISA Transactions*, 66:233–240, 2016.

[6] H.J. Jayakrishnan. Position and attitude control of a quadrotor uav using super twisting sliding mode. *IFAC-PapersOnLine*, 49(1):284–289, 2016.

[7] K. Chang, Y. Xia, K. Huang, and D. Ma. Obstacle avoidance and active disturbance rejection control for a quadrotor. *Neurocomputing*, 190:60–69, 2016.

[8] Y. Yang and Y. Yan. Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control. *Aerospace Science and Technology*, 54:208–217, 2016.

[9] R. Perez-Alcocer, J. Moreno-Valenzuela, and R. Miranda-Colorado. A robust approach for trajectory tracking control of a quadrotor with experimental validation. *ISA Transactions*, 65:262–274, 2016.

[10] D. Cabecinhas, R. Cunha, and C. Silvestre. A nonlinear quadrotor trajectory tracking controller with disturbance rejection. *Control Engineering Practice*, 26:1–10, 2014.

[11] A. Modirrousta and M. Khodabandeh. A novel hybrid controller design for an uncertain quadrotor with disturbances. *Aerospace Science and Technology*, 45:294–308, 2015.

[12] W. Dong, G.-Y. Gu, X. Zhu, and H. Ding. A high-performance flight control approach for quadrotors using a modified active disturbance rejection technique. *Robotics and Autonomous Systems*, 83:177–187, 2016.

[13] F. Munoz, I. Gonzalez-Hernandez, S. Salazar, E.S. Espinoza, and R. Lozano. Second order sliding mode controllers for altitude control of a quadrotor uas: Real-time implementation in quadrotor environments. *Neurocomputing*, 233:61–71, 2016.

[14] A.T. Azar and Q. Zhu. *Advances and applications in sliding mode control systems*. Springer, Switzerland, 2015.

[15] J.A. Moreno. On strict lyapunov functions for some non-homogeneous super-twisting algorithms. *Journal of the Franklin Institute*, 351(4):1902–1919, 2014.

[16] B. Sumantri, N. Uchiyama, and S. Sano. Least square based sliding mode control for a quad-rotor helicopter and energy saving by chattering reduction. *Mechanical Systems and Signal Processing*, 66–67:769–784, 2016.

[17] S.-B. Choi, D.-W. Park, and S. Jayasuriya. A time-varying sliding surface for fast and robust tracking control of second-order uncertain systems. *Automatica*, 30(5):899–904, 1994.

[18] S. Li, Y. Wang, J. Tan, and Y. Zheng. Adaptive rbfnns/integral sliding mode control for a quadrotor aircraft. *Neurocomputing*, 216:126–134, 2016.

[19] S. T. Venkataraman and S. Gulati. Control of nonlinear systems using terminal sliding modes. *Journal of Dynamic Systems, Measurement, and Control*, 115(3):554–560, 1993.

[20] Y. Feng, X. Yu, and Z. Man. Non-singular terminal sliding mode control of rigid manipulators. *Automatica*, 38(12):2159–2167, 2002.

Controller Tuning Strategy for Quadrotor MAV Carrying a Cable-suspended Load

Nestor A. Santos Ortiz*, Edouard Laroche†, Renaud Kiefer‡ and Sylvain Durand§
ICube UMR 7357, 300 bd. Sébastien Brant - CS 10413 - F-67412 Illkirch Cedex, France

ABSTRACT

This paper presents a controller tuning strategy for a quadrotor MAV carrying a cable-suspended load. In our study, no measurement nor estimation of the load's position is used in the control strategy and only the quadrotor attitude and position are controlled. The tuning of the controllers has been done in order to satisfy mixed H_∞ requirements and a pole-location requirement. The resolution is made with an available tool based on non-smooth optimization. The proposed methodology, allowing to find a good trade-off between fast displacements of the MAV and well damped oscillations of the load, is validated in simulation.

Keywords : PID control, quadrotor, suspended load perturbations, fixed-structure controller tuning.

1 INTRODUCTION

Multirotor micro air vehicles (MAV) are now considered for multiple kinds of missions such as search and rescue [1], surveillance [2], exploration, field recognition and cargo transportation [3]. This project focuses on the payload transport missions, specifically on the problem of carrying a load suspended with a cable. Over the last few years, the increasing demand of this capability on quadrotors has motivated researchers to study this problem and diverse efforts have already been done to solve it. Different control techniques have been already implemented for quadrotor applications with the aim to improve their performances and capabilities.

In general, pioneers on the subject started with the design of the control strategies of quadrotors. Altug *et al.* [4] proposed the use of backstepping controllers for position control considering the dependency with the attitude control. Hamel *et al.* [5] developed control Lyapunov functions by separating the rigid body dynamics and the motors dynamics. They also introduced the use of quaternions for attitude error estimation. Pounds *et al.* presented the design of a controller based on two cascaded loops integrating a double lead compensator and a pure proportional feedback loop [6].

*Email address: nestor-alonso.santos-ortiz@insa-strasbourg.fr
†Email address: laroche@unistra.fr
‡Email address: renaud.kiefer@insa-strasbourg.fr
§Email address: sylvain.durand@insa-strasbourg.fr

Returning to the specific problem here discussed, it can be viewed and analyzed from different angles. One approach focuses on the control of the suspended load so it follows a specific trajectory while damping the oscillations. Faust *et al.* developed an algorithm based on machine learning for the generation of trajectories reducing the swing of the load [7]. They, as most of other researchers, use the model of a rigid 3D pendulum for the cable-suspended load. Dai *et al.* preferred to use a more accurate model considering the cable as a series of connected links and they design a controller implementation to align these links vertically while transporting the load to a desired position [8]. In their results, an adaptive controller is used to compensate the payload's mass uncertainty that demonstrates better performances than a PID controller with constant gains. Other design methods have been proposed in order to reject the perturbations generated by the oscillations of the suspended load seen as a disturbance. For example, in [9, 10], nonlinear controllers are presented for trajectory tracking based on partial feedback linearization and on the backstepping technique, respectively. In [11], both disturbance rejection and swing reduction problems are considered. A nonlinear H_∞ controller added to a control law obtained by the Lyapunov redesign technique is proposed and evaluated in simulation. Finally, a third variant of the problem considers the case of a load carried by multiple cooperative MAVs [12].

In most of the cases, it is considered that the complete state vector is measurable. For this reason, experimentation is frequently done in rooms equipped with motion capture vision systems. A problem that arises here is the implementation of these control strategies for outdoor flights. In this paper, the classical PID control strategy is implemented on a quadrotor where no sensor is employed to estimate the orientation of the cable nor the load's position. The PID structure here implemented is similar to the B-type presented in [13] by Szafranski where derivation is applied directly on the measurement signal and not on the error signal. The PID controller has been commonly used on MAVs and, as in many other applications, it is frequently included in studies where different controllers are to be evaluated [14]. Moreover, this kind of controller is commonly implemented in open source and commercial autopilot frameworks used for MAV. The analysis of the controller performance is presented with a tuning strategy for the gain parameters with optimization algorithms considering the dynamic model of the load in the system.

In Section 2, the complete system modeling is detailed. Section 3 details the attitude control strategy, including simulations results. Section 4 presents an analysis for the position control. Section 6 concludes the paper and discusses some future work.

2 System Modeling

After introducing the quadrotor MAV structure used for the present study, the modeling of the system with the suspended load is presented using the Euler-Lagrange formalism. The suspended load is considered as a pendulum with the cable assumed to be a rigid massless link attached at a certain distance under the center of gravity of the quadrotor.

2.1 Millenium quadrotor

The *Millenium* quadrotor, developed at INSA Strasbourg, is considered in this project [15]. Its structure is entirely made with composite materials that offer excellent mechanical properties while minimizing the weight, increasing the payload that it can carry up-to 1 kg. Its configuration is of X type, however its structure has only a longitudinal symmetry. The two frontal arms are more open so they do not appear in the field of view of a camera looking towards that direction, see Figure 1.

A Pixhawk Autopilot board is used for the control of this drone which runs the PX4 firmware pilot. PID type controllers are used by default and their parameters can be easily modified via a ground control station.

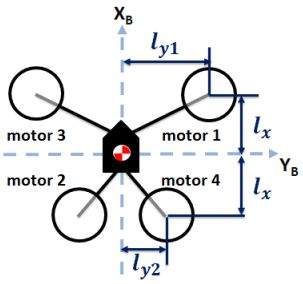


Figure 1: INSA's quadrotor's framework and motors map.

2.2 Kinematics and system coordinates

Two main reference frames are used for modeling the MAV: the inertial frame attached to a point in the earth ($\mathscr{R}_E = \boldsymbol{X_E}, \boldsymbol{Y_E}, \boldsymbol{Z_E}$) and the body fixed frame ($\mathscr{R}_B = \boldsymbol{X_B}, \boldsymbol{Y_B}, \boldsymbol{Z_B}$). Inertial frame is defined as NED type since it is the case in the PX4 board controller.

Position vector $\boldsymbol{\xi} \in \mathscr{R}_E$ is defined as the position of the body frame origin in the inertial frame coordinate system.

$$\boldsymbol{\xi} = \begin{bmatrix} x_E & y_E & z_E \end{bmatrix}^T \tag{1}$$

Euler angles are used to define the attitude of the quadrotor with respect to the inertial frame in the following order: yaw (ψ) around the Z-axis, then pitch (θ) around Y-axis and finally roll (φ) around the X-axis.

$$\boldsymbol{\eta} = \begin{bmatrix} \varphi & \theta & \psi \end{bmatrix}^T \tag{2}$$

The rotation matrix $\boldsymbol{R_E^B}$ from body to inertial frame is defined in (3).

$$\boldsymbol{R_B^E} = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\varphi - s_\psi c_\varphi & c_\psi s_\theta c_\varphi + s_\psi s_\varphi \\ s_\psi c_\theta & s_\psi s_\theta s_\varphi + c_\psi c_\varphi & s_\psi s_\theta c_\varphi - c_\psi s_\varphi \\ -s_\theta & c_\theta s_\varphi & c_\theta c_\varphi \end{bmatrix} \tag{3}$$

To transform the derivatives of the Euler angles from the inertial frame, $\dot{\boldsymbol{\eta}}$, to the body frame, $\boldsymbol{\omega_B}$, the transformation matrix $\boldsymbol{J}$ defined in Equation (4) is used.

$$\boldsymbol{\omega_B} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \boldsymbol{J} \cdot \dot{\boldsymbol{\eta}} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\varphi & s_\varphi c_\theta \\ 0 & -s_\varphi & c_\varphi c_\theta \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{4}$$

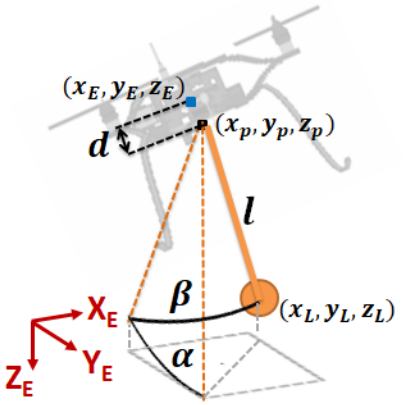


Figure 2: Suspended load parametrization.

For the load displacement analysis, a new coordinate frame fixed to the pendulum ($\mathscr{R}_P = \boldsymbol{X_P}, \boldsymbol{Y_P}, \boldsymbol{Z_P}$) is defined with its origin on the attachment point and with $\boldsymbol{Z_P}$ coincident with the cable. The rotation matrix $\boldsymbol{R_P^E}$ from the pendulum frame to the inertial frame is defined with angles α and β corresponding to rotation of the pendulum firstly around the $\boldsymbol{X_E}$ axis and then around the already rotated $\boldsymbol{Y_E'}$ axis, as illustrated in Figure 2.

$$\boldsymbol{R_P^E} = \begin{bmatrix} c_\beta & 0 & s_\beta \\ s_\alpha s_\beta & c_\alpha & -s_\alpha c_\beta \\ -c_\alpha s_\beta & s_\alpha & c_\alpha c_\beta \end{bmatrix} \tag{5}$$

In addition, the position of the cable fixation point is under a distance d from the center of gravity of the quadrotor which is chosen as the body frame origin. Finally, the position vector of the load in the inertial frame, $\boldsymbol{\xi_L}$, is obtained in Equation (6) with l the length of the cable.

$$\boldsymbol{\xi_L} = \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} = \boldsymbol{\xi} + \boldsymbol{R_B^E} \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} + \boldsymbol{R_L^E} \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix} \quad (6)$$

2.3 Forces and torques

The quadcopter is an under-actuated system since only four propellers induce the movement while it has a total of 6 degrees of freedom. In Figure 3, the forces and moments induced by the propellers are illustrated. They are considered to be proportional to the square of the propeller rotation speed:

$$\begin{cases} F_i = k_f \cdot \omega_i^2 \\ \tau_i = k_c \cdot \omega_i^2 \end{cases} \quad (7)$$

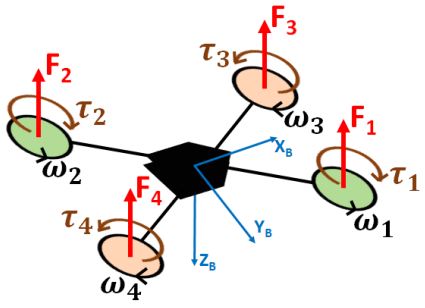


Figure 3: Forces and moments induced by the propellers.

To simplify the analysis, a change among the input variables is defined as following: the control inputs are defined as the total thrust, T, and the torques, τ_φ, τ_θ and τ_ψ, with respect to the three body frame axis. These new inputs variables depend on the rotation speed of propellers, ω_i, and are calculated with Equation (8). The matrix used in this equation is defined considering the particular structure geometry of the quadrotor of this study depicted in Figure 1:

$$\begin{bmatrix} T \\ \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} k_f & k_f & k_f & k_f \\ -k_f l_{y1} & k_f l_{y2} & k_f l_{y1} & -k_f l_{y2} \\ k_f l_x & -k_f l_x & k_f l_x & -k_f l_x \\ k_c & k_c & -k_c & -k_c \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (8)$$

2.4 Dynamic model

Euler-Lagrange equations are used to obtain the dynamic model of the complete system : quadrotor with suspended load. The translational kinetic energy is composed by both the quadrotor and the load energies :

$$K_{translation} = \frac{1}{2}\left(m\dot{\boldsymbol{\xi}}^T\dot{\boldsymbol{\xi}} + m_L\dot{\boldsymbol{\xi}_L}^T\dot{\boldsymbol{\xi}_L}\right) \quad (9)$$

where m is the quadrotor's mass and m_L is the load's mass. On the other hand, considering the load as a punctual mass, its rotational kinetic energy is neglected and only the quadrotor energy is included :

$$K_{rotation} = \frac{1}{2}\boldsymbol{\omega_B}^T\boldsymbol{I}\boldsymbol{\omega_B} \quad (10)$$

where I is the inertial matrix of the quadrotor with respect to the body frame and which is considered as a diagonal one ($I = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz})$) since the MAV principal axes are supposed to be coincident with the body frame axes for simplification. However, in reality they are slightly rotated.

Finally, the potential energy is defined in Equation (11) where g represents the gravity constant. The negative sign of this equation is due to the positive direction of the Z axis which is downwards.

$$U = -m_D g\boldsymbol{Z_E}^T\boldsymbol{\xi} - m_L g\boldsymbol{Z_E}^T\boldsymbol{\xi_L} \quad (11)$$

The dynamic of the system is summarized with the Lagrangian L equal to the kinetic energy minus the potential energy.

$$L = K_{translation} + K_{rotation} - U \quad (12)$$

Then, the Euler-Lagrange equations represents the equations of motion of the system for the generalized coordinates $\boldsymbol{q} = \begin{bmatrix} x_E & y_E & z_E & \varphi & \theta & \psi & \alpha & \beta \end{bmatrix}^T$:

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{q}_i}\right] - \frac{\partial L}{\partial q_i} = Q_i, \quad i = 1, ..., 8 \quad (13)$$

The generalized force vector $\boldsymbol{Q}$ is defined by the input force $\boldsymbol{F}$, the input torque $\boldsymbol{\tau}$ and the gyroscopic effect moment of the rotors $\boldsymbol{\Gamma}$. Note that the frictional forces due to air resistance have been neglected.

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{\tau} - \boldsymbol{\Gamma} \\ \boldsymbol{0_{2\times 1}} \end{bmatrix} \quad (14)$$

$$\boldsymbol{F} = \boldsymbol{R_B^E}\begin{bmatrix} 0 & 0 & -T \end{bmatrix}^T \quad (15)$$

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_\varphi & \tau_\theta & \tau_\psi \end{bmatrix}^T \quad (16)$$

The gyroscopic moments of the rotors are calculated using Equation (17) with $\Omega = \omega_1 + \omega_2 - \omega_3 - \omega_4$ and I_r correspond to the angular moment of rotors.

$$\boldsymbol{\Gamma} = I_r\left(\boldsymbol{\omega_B} \times \boldsymbol{Z_B}\right)\Omega = I_r \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Omega = \begin{bmatrix} I_r q\Omega \\ -I_r p\Omega \\ 0 \end{bmatrix} \quad (17)$$

Finally using the matrix representation of the Euler-Lagrange equations, motion equation can be written as follows.

$$\boldsymbol{M(q)\ddot{q}} + \boldsymbol{C(q,\dot{q})\dot{q}} + \boldsymbol{G(q)} = \boldsymbol{Q} \quad (18)$$

$$\boldsymbol{\ddot{q}} = \boldsymbol{M(q)}^{-1}\left(\boldsymbol{Q} - \boldsymbol{C(q,\dot{q})\dot{q}} - \boldsymbol{G(q)}\right) \quad (19)$$

Matrices $\boldsymbol{M(q)}$, $\boldsymbol{C(q,\dot{q})}$ and $\boldsymbol{G(q)}$ have been easily obtained with a Matlab script but they are very extensive to appear in this paper.

2.5 *Propellers dynamic model identification*

Dynamic model of the motor-propeller set is commonly identified by a first-order model [16], with a time constant τ and a gain K in addition to a delay d due to the period of the pulse-width modulated input signal to the motors ESC.

$$T(s) = e^{-d \cdot s} \frac{K}{1 + \tau \cdot s} \tag{20}$$

Identification experimental data showed that the time constant is different when the motor speed is increased and when it is lowered. In addition, the static gain identified follows a quadratic behavior in function of the positive pulse duration of the PWM input of ESC controllers which varies from 1.1 ms to 1.9 ms with a frequency of 400Hz. Then, the non linear model in Equation (21) has been defined, with y the rotation speed in RPM and u the ESC input in ms.

$$\begin{aligned} &K = a_2 u^2 + a_1 u + a_0 \\ &\begin{cases} \dot{y} = \frac{1}{\tau_1}(Ku - y) & , Ku - y \geq 0 \\ \dot{y} = \frac{1}{\tau_2}(Ku - y) & , Ku - y < 0 \end{cases} \end{aligned} \tag{21}$$

A noticeable improvement fit has been observed with the proposed nonlinear model compared to the linear model.

2.6 *System parameters*

The dimensions and mass parameters of the quadrotor have been directly measured. Furthermore, the diagonal inertia matrix values have been estimated with the compound pendulum method described in [17]. The identified parameter values are presented in Table 1 and compared to the values obtained with the CAD model of the system. As observed, the CAD model values help as an approximation to validate the coherency of the measurements.

Table 1: Dimensions, mass and inertia parameters.

DATA	CAD	Measurements
m $[kg]$	2.834	2.832 ± 0.02
l_x $[m]$	0.262	0.270 ± 0.005
l_{y1} $[m]$	0.284	0.295 ± 0.005
l_{y2} $[m]$	0.222	0.225 ± 0.005
I_{xx} $[kg \cdot m^2]$	0.064	[0.054 - 0.061]
I_{yy} $[kg \cdot m^2]$	0.077	[0.060 - 0.068]
I_{zz} $[kg \cdot m^2]$	0.127	[0.107 - 0.118]

Thrust coefficient has been approximated from experimental flight data where the quadrotor stayed on a hover position. Drag coefficient has not been identified and an approximated value has been chosen based on online databases.

3 Attitude Control

As already mentioned before, the interest of this paper is not to design a better controller structure for this application, but to suggest an original strategy to calculate the best parameters of the imposed controllers which structure is defined in Figure 4. It is a pure proportional controller in the outer-loop cascaded with a PID in the inner-loop.

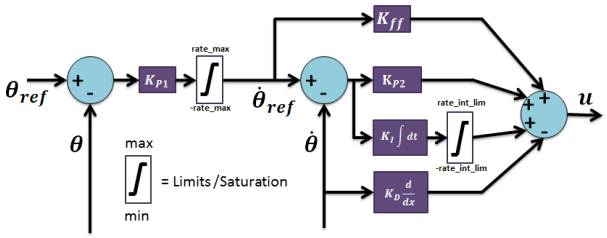


Figure 4: PX4 attitude PID controller structure.

3.1 *Tuning of the control parameters*

For the tuning of the controller parameters, a solution with optimization algorithms for non-smooth and non-convex problems is available (see [18, 19]). The optimization problem solved by this algorithm, considering only one plant model, is of the form

$$\begin{aligned} &\underset{p \in \Re}{\text{minimize}} \ \max_{i=1,\ldots,n_f} \{f_i(p)\} \\ &\text{subject to} \ \max_{j=1,\ldots,n_g} \{g_j(p)\} \leq 1 \end{aligned} \tag{22}$$

where p is the vector of tunable parameters. Functions f_i and g_j represent the requirements, considered as soft and hard respectively. There can be as many requirements as desired in the form of H_2 or H_∞ norms of weighted transfer functions, or in terms of a pole location constrain. Values γ_s and γ_h represent the optimization results for the soft and hard requirements.

Moreover, this tuning method can be easily implemented in Matlab with the *'systune'* tool. Here, this has been used looking to satisfy three different hard performance requirements (same results are obtained with only soft requirements):

Req 1: Maximum tracking error from the angle reference to the angle measurement in the frequency domain;

Req 2: Maximum gain from disturbance to angle measurement in the frequency domain;

Req 3: Pole location of the closed-loop system.

To simplify the problem, the dynamic model for each of the controlled variables has been separated and simplified. Two different models have been used for roll and pitch, the

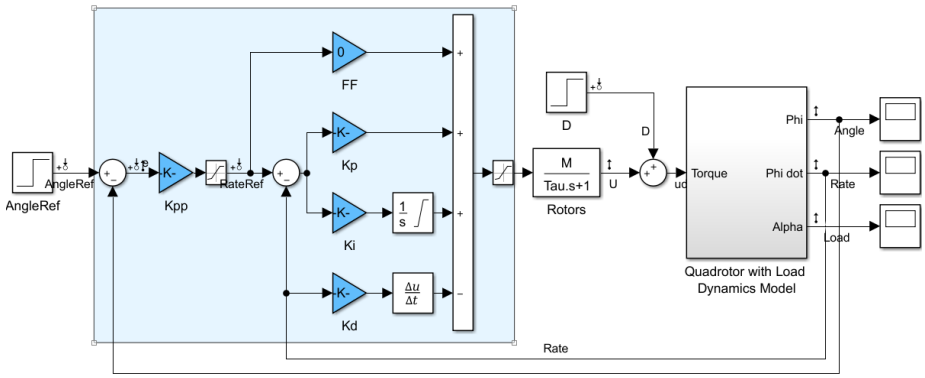


Figure 5: Simplified control system for roll (*KA2*).

first one without considering the pendulum load, and the second one including it. This way two different sets of parameters for attitude controllers have been obtained and evaluated. The following notations are used for next references:

***KA1*:** Tuning of the parameters when considering the model of the system without the load;

***KA2*:** Tuning of the parameters when considering the model of the system with the load.

Figure 5 shows the general aspect of the simulink model used for tuning the roll controller in case *KA2*. A linear model has been used for the actuators with a gain M from the control output to the system input due to the PX4 normalized mixer (mixer transforms the controller outputs into the motors ESC control inputs).

For both cases, the weighting functions used on the performance channel have been adjusted iteratively by looking for the most demanding performances that could be satisfied ($\gamma_h \leq 1$). In a first step, the bandwidth, the reject of perturbation, the damping ratio and the decay ratio (associated to the phase and the real part of the dominant poles respectively) of dominant poles were required to a relative high value for this specific application. The system could never satisfy these constrains but this gave an intuition of the performances limits of the system. In a next step, the requirements have been lowered until all of them could be satisfied. Figures 6(a) and 6(b) show the evaluation of the final requirements for cases *KA1* and *KA2* respectively.

It can be observed that in the case of *KA1*, the bandwidth of the system is a little higher than in the case of *KA2*. Note that when a higher bandwidth was requested in the case of *KA2*, it could not be satisfied without affecting the stability of the system (reducing the damping ratio and decay component for the pole location constrain).

3.2 Simulation results

To evaluate roll and pitch controllers, two different tests have been performed in simulation.

In the first test, the MAV is controlled to be on the hover state, and the load is initially perturbed. Specifically, the load is initialized with angles α and β of $20°$. Figure 7 shows the results for the control of roll attitude, for both set of controller parameters *KA1* and *KA2*. The response for pitch is almost the same as the obtained for roll. As it can be clearly observed, with the *KA2* parameters, the load oscillations are rapidly attenuated, contrarily to the case with parameters *KA1*.

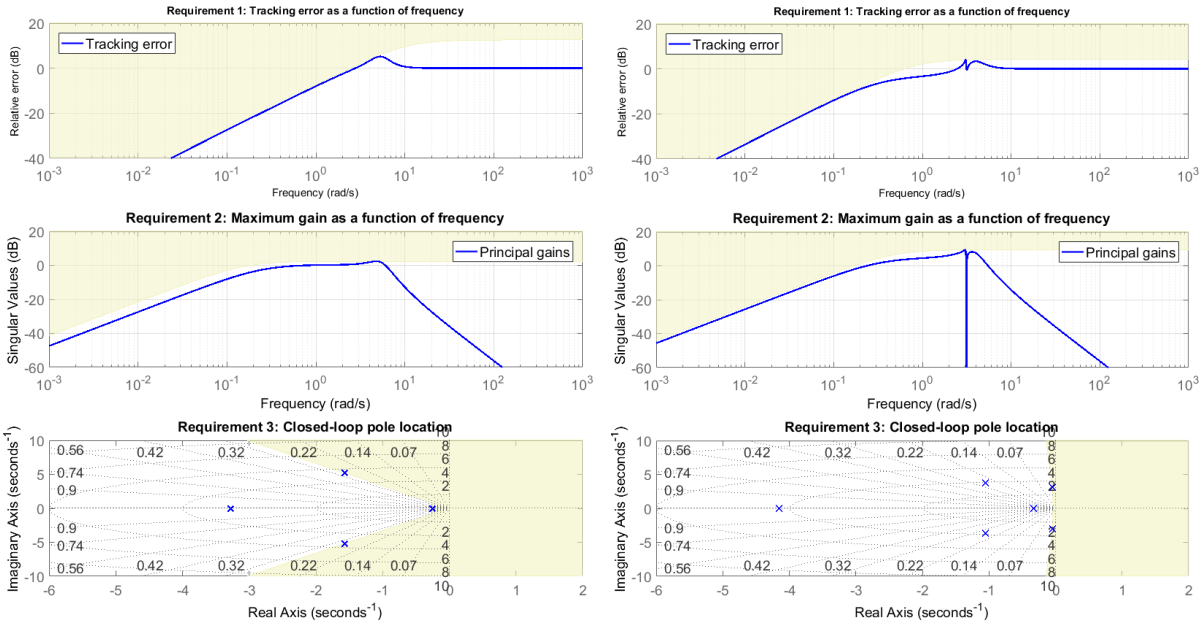
In a second test, the MAV attitude angles ϕ and θ were initially set to $20°$, and the control reference is the horizontal attitude. This time the load is initially at rest but it will be excited by the horizontal movement induced with the initial attitude of the MAV. For this test, results are presented in Figure 8. Again, with the controllers *KA2*, the oscillations of the load are more rapidly attenuated while the reference attitude is also reached.

These results already demonstrate that good stability performances can be achieved considering the model of the load in the tuning process, even when no feedback of the load position is used in the control strategy.

4 Position control

The same controller structure as used for attitude in section 3 is employed for the position control. Note that for horizontal movements, the position controller is cascaded with attitude controller. The X, Y and Z control signals define a reference thrust vector in the inertial frame which orientation is used to define the pitch and roll reference angles. Besides, in function of the current attitude of the drone, the thrust vector is redefined and its magnitude is calculated for the control of the altitude. As a result, control is done for X, Y, Z, and yaw. Again, the same process followed for the attitude controllers tuning has been implemented in this case. Two models have been also used :

***KP1*:** Tuning of the parameters when considering the model of the system without the load and *KA1* attitude controllers;



(a) KA1. (b) KA2.

Figure 6: Requirements for roll controller tuning.

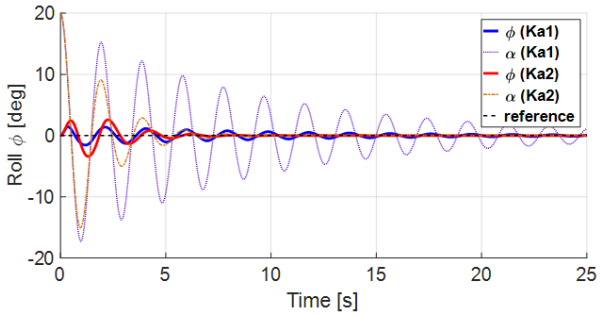


Figure 7: Quadrotor's roll attitude in time, first test.

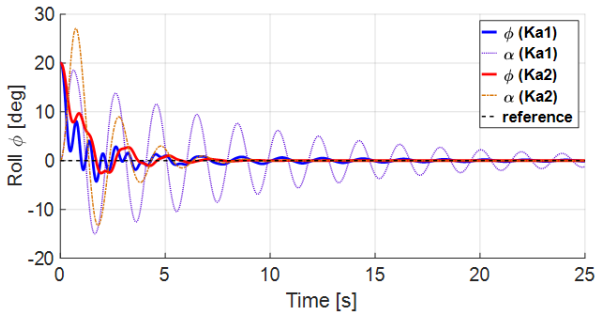


Figure 8: Quadrotor's roll attitude in time, second test.

trollers.

This time, only one test has been used for evaluation. From an initial position at rest, the test consisted in step variations of the reference signal. Note that, rather than using a trajectory generation algorithm as it is usually the case, steps have been used to fully excite the system and emphasize its stability properties. The reference positions formed a square trajectory with the center in the initial position. A 3D visualization of thethis test appears on Figure 9.

The comparison of the reference tracking results with the different sets of parameters can be appreciated in Figure 10. On the one hand, with parameters *KP1-KA1*, Figure 10(a), the system shows relative good performances. The reference is rapidly reached with a small overshot and few oscillations. On the other hand, with parameters *KP2-KA2*, Figure 10(b), the system is less dynamic. However, in terms of the stabilization of the load, better results are obtained with the later parameters. As it can be observed in Figure 11, with the second set of controller parameters, the oscillations of the load are more attenuated and the highest amplitude is around four times lower.

5 Discussion on the tuning strategy

From our experience in the tunning process, we draw several recommendations.

***KP2*:** Tuning of the parameters when considering the model of the system with the load and *KA2* attitude con-

Model with or without the load. It is possible to use a synthesis model that does not include the load, which simplifies

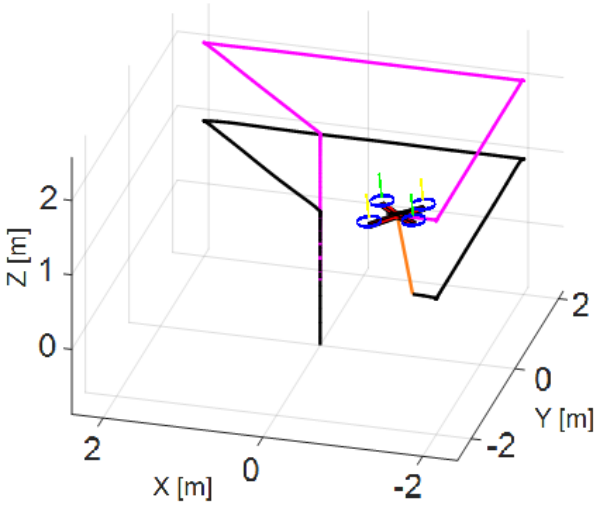
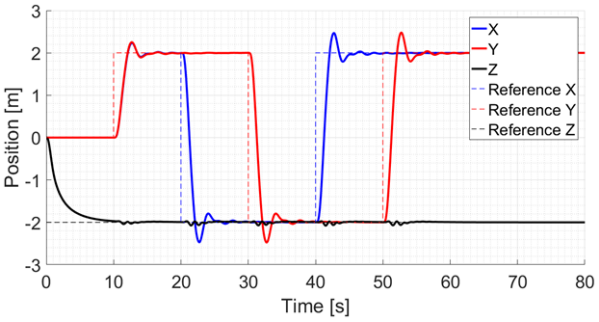
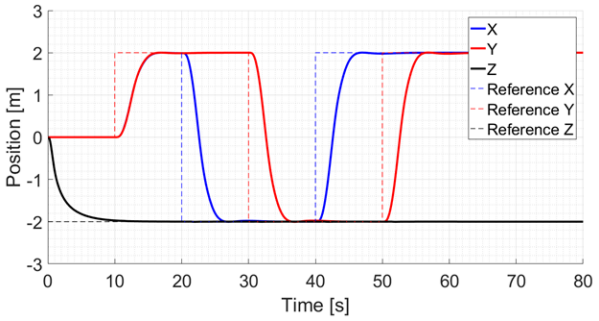


Figure 9: 3D visualization of the position control simulation (with controller *KP2* and *KA2*)



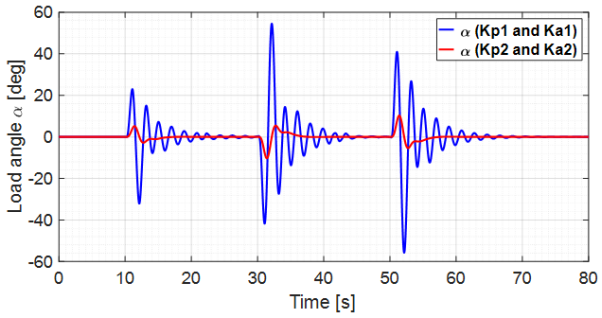
(a) Controller *KP1* and *KA1*.



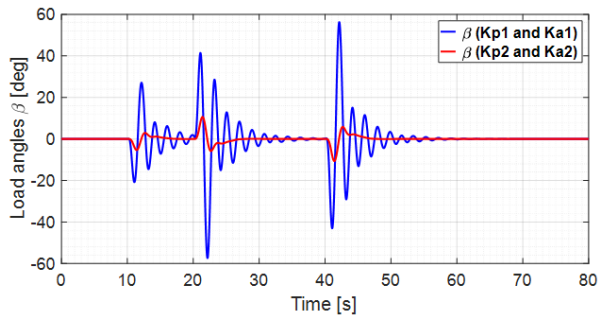
(b) Controller *KP2* and *KA2*.

Figure 10: Quadrotor's position with respect to time.

the modeling step. With this model, it is possible to obtain a much faster control of the orientation and the position of the UAV, but at the cost of undamped or unstable load oscillations if the bandwidth is chosen higher than the load oscillation frequency. The model including the load displacements was found convenient to tune a controller that damps the load oscillations.



(a) Angle α.



(b) Angle β.

Figure 11: Load angle α and β during the position control test.

Pole location requirement only. In an attempt to simplify the requirements specification, the pole location constrain has been used as the only requirement. However, we observed that the resulting dynamics of the UAV is degraded and then sticks to the dynamics of the load.

Remove the pole location requirement. We replaced the pole location by a frequency requirement on the transfer between the reference and the load displacement, which results in purely frequency-domain requirements. Even if these requirements allowed to damp the load oscillations, we obtained a better results by including the pole placement requirements. Notice that once the pole placement requirement is included, the requirements on the transfer between the reference and the load displacement can be removed without any drawback.

Hard vs soft requirements. The possibility of prioritizing some of the requirements (using hard and soft requirements) has also been investigated. However, better results have been obtained by keeping the same strength for all requirements.

6 Conclusions and Future Work

The problem of the control of a quadrotor MAV with a cable-suspended load has been studied. The modeling of the system including the load has been described. It has been

considered that the attachment point of the cable is not coincident with the center of gravity of the MAV, allowing the load to induce perturbations on the drone's attitude. Moreover, a tuning methodology for the controllers has been proposed in order to find a good trade-off between the response time of the vehicle and the damping of the load oscillations, even when there is no feedback of them. The validity of this methodology has been proved with simulations. This can be useful during outdoor flights with standard quadrotors, avoiding the need of adding sensors for the load position estimation and maintaining the simplicity of this type of control implementation. In future work, the evaluation of the tuned controllers will be performed experimentally using the *Millenium* quadrotor from INSA Strasbourg to observe the robustness against non modeled dynamics and uncertainty on the identified parameters of the system.

ACKNOWLEDGEMENTS

This project is supported by the *CoreDrone* project, an internal project from ICube Laboratory, and INSA Strasbourg.

REFERENCES

[1] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka. Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. *IEEE Robotics & automation magazine*, 19(3):46–56, 2012.

[2] H. C. Yang, R. AbouSleiman, B. Sababha, E. Gjioni, D. Korff, and O. Rawashdeh. Implementation of an autonomous surveillance quadrotor system. In *AIAA Infotech Aerospace Conference*, page 2047, 2009.

[3] R. Mileham. Prime movers. *Engineering & Technology*, 11(4):70–72, 2016.

[4] E. Altug, J. P. Ostrowski, and R. Mahony. Control of a quadrotor helicopter using visual feedback. In *IEEE Int. Conf. Robotics and Automation*, pages 72–77 vol.1, 2002.

[5] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski. Dynamic modelling and configuration stabilization for an x4-flyer. *IFAC Proceedings Volumes*, 35(1):217–222, 2002.

[6] P. Pounds, R. Mahony, P. Hynes, and J. M. Roberts. Design of a four-rotor aerial robot. In *Australasian Conference on Robotics and Automation (ACRA)*, pages 145–150. Australian Robotics & Automation Association, 2002.

[7] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia. Learning swing-free trajectories for uavs with a suspended load. In *IEEE Int. Conf. Robotics and Automation*, pages 4902–4909, May 2013.

[8] S. Dai, T. Lee, and D. S. Bernstein. Adaptive control of a quadrotor UAV transporting a cable-suspended load with unknown mass. In *IEEE Conf. Decision and Control*, pages 6149–6154, December 2014.

[9] I. H. Beloti Pizetta, A. S. Brando, and M. Sarcinelli-Filho. Modelling and control of a quadrotor carrying a suspended load. In *Education and Development of Unmanned Aerial Systems (RED-UAS) 2015 Workshop Research*, pages 249–257, November 2015.

[10] K. Klausen, T. I. Fossen, and T. A. Johansen. Nonlinear control of a multirotor UAV with suspended load. In *Int. Conf. Unmanned Aircraft Systems (ICUAS)*, pages 176–184, June 2015.

[11] Guilherme V Raffo and Marcelino M de Almeida. Nonlinear robust control of a quadrotor uav for load transportation with swing improvement. In *American Control Conference (ACC), 2016*, pages 3156–3162. IEEE, 2016.

[12] K. Sreenath and V. Kumar. Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots. June 2013.

[13] G. Szafranski and R. Czyba. Different approaches of pid control uav type quadrotor. *International Micro Air Vehicles (IMAV)*, 2011.

[14] A. Rodic, G. Mester, and I. Stojkovic. Qualitative evaluation of flight controller performances for autonomous quadrotors. In *Intelligent Systems: Models and Applications*. Springer, January 2013.

[15] R. Kiefer and M. Vedrines. Proposition of design method and optimization of multirotors composite structures. In *International Micro Air Vehicles (IMAV)*, Aachen, Germany, September 2015.

[16] E. Roussel, V. Gassmann, and E. Laroche. Modelling and identification of a coaxial birotor UAV from scarce flight data. In *Proc. European Control Conf. (ECC)*, pages 2158–2164, June 2016.

[17] H. A. Soule and M. P. Miller. The experimental determination of the moments of inertia of airplanes. Technical Report 467, National Advisory Committee for Aeronautics, Langley Field, VA, USA, 1934.

[18] P. Apkarian, P. Gahinet, and C. Buhr. Multi-model, multi-objective tuning of fixed-structure controllers. In *European Control Conference (ECC)*, pages 856–861. IEEE, 2014.

[19] P. Apkarian and D. Noll. Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control*, 51(1):71–86, 2006.

Technical paper

Landing and Take-off on/from Sloped and Non-planar Surfaces with more than 50 Degrees of Inclination

M. Tognon* and A. Franchi
LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France

ABSTRACT

This technical paper summarizes the recent experimental results concerning the challenging problem of landing and take-off on/from a sloped surface with an aerial vehicle exploiting the force provided by an anchored taut tether. A special regard is given to the practical aspects concerning the experimental part. In this manuscript we show extreme landing and take-off maneuvers on slopes with at least 50° inclination and non flat surfaces, such as, e.g., on industrial pipes.

1 INTRODUCTION

The popularity of aerial vehicles is growing day by day thanks to their versatility. Indeed they are currently used for several applications ranging from civil inspection to search and rescue. Those kinds of applications often require to land the vehicles on a sloped surface. For example, in the context of the European project *Aeroarms*[1], the aerial robot has to deploy a magnetic crawler or a sensor on industrial pipes that are often non-horizontal. Figure 1 shows an example of a tethered landing on a pipe tilted by an angle of $60°$.

Take-off and landing from/on a sloped surface requires to control both position and orientation (pitch) independently, since the last has match the inclination of the surface. This makes the task very challenging for a standard Vertical Take-off and Landing vehicle (VTOL) in a free-flight configuration. Indeed due to underactuation, VTOLs can control only the position and the yaw angle, while roll and pitch are a byproduct of the vehicle acceleration and the gravity force. The method normally used exploits the flatness of the system to plan a desired trajectory that ends in the landing spot and with the proper orientation [1]. However, the landing can not be done in a stable and safe configuration, i.e., with zero velocities and accelerations. Instead it requires an agile maneuver that has to be executed with very high precision. Indeed small tracking errors can easily lead to misses of the landing surface or to crashes. Furthermore, adhesive membranes are used to enforce the cohesion between the vehicle and the surface to reduce the chances to fly away after the contact. Those facts makes the task very complex and prone to errors and failures.

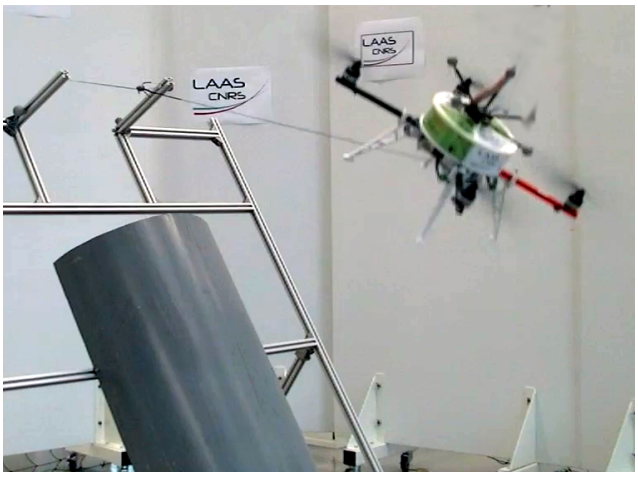


Figure 1: Tethered landing on a sloped pipe tilted by $60°$.

On the other hand, a much more reliable method gaining interest, consists on the use of a tether that connects the vehicle to a fixed point on the surface (see Fig. 1). Recent works have already studied this system, analyzing its principal properties as differential flatness, controllability and observability in the 2D case [2, 3, 4] and also in the 3D cases [5, 6, 7]. The tether, with the help of an actuated winch, has been also used in [5, 8] to land an underactuated aerial vehicle on a moving platform. Instead, in [6] we have shown that a passive tether (with no extra actuation) is enough to accomplish landing and take-off on/from sloped surfaces in a robust and reliable way.

In this paper we shall thoroughly discuss all the technical and practical aspects concerning the real execution of the landing and take-off maneuvers exploiting a tether. Here we consider a quadrotor-like vehicle and sloped surfaces tilted by at least $50°$.

2 TETHERED LANDING

In [6] we proved that the tether configuration and the inclination of the robot w.r.t. the cable are flat outputs of the system. This means that the two quantities can be precisely

*Email address(es): marco.tognon@laas.fr
[1] http://www.aeroarms-project.eu/

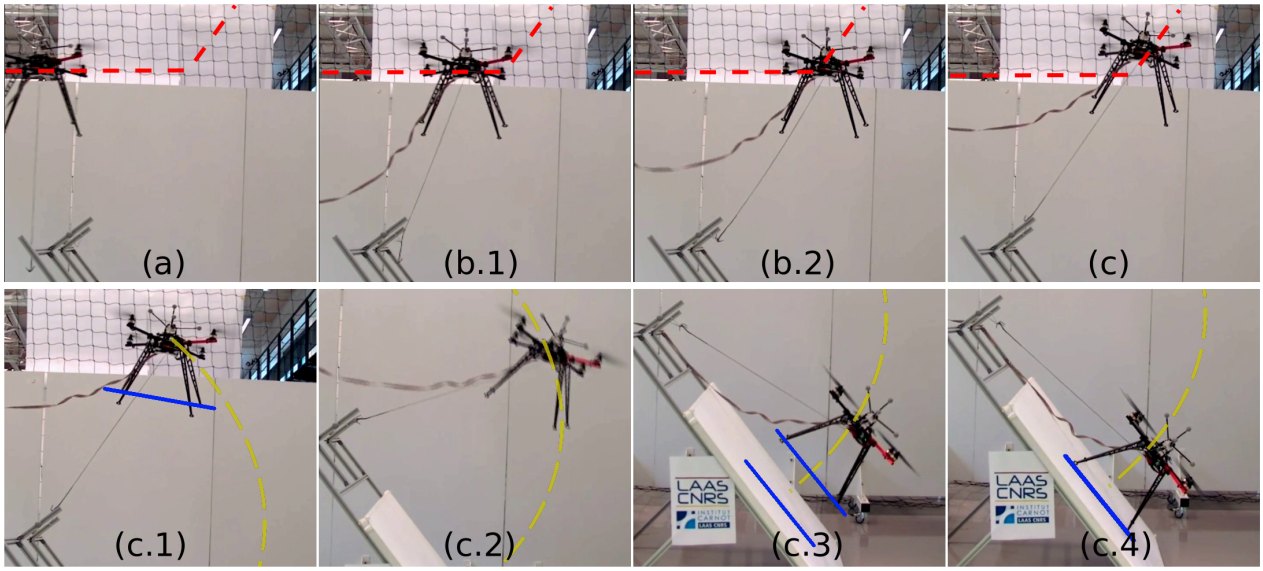


Figure 3: Sequence of images of a real experiment with a sloped surface tilted by 50°. The first row of images represents the experimental part in which the quadrotor is in a free-flight condition. In this case a standard position controller is used to track the desired position trajectory marked with a dashed red line. The second row of images represents the experimental part in which the quadrotor is tethered to the surface. In this case the controller proposed in [6] is used to track the desired position and attitude trajectories marked with a dashed yellow line and a solid blue line, respectively.

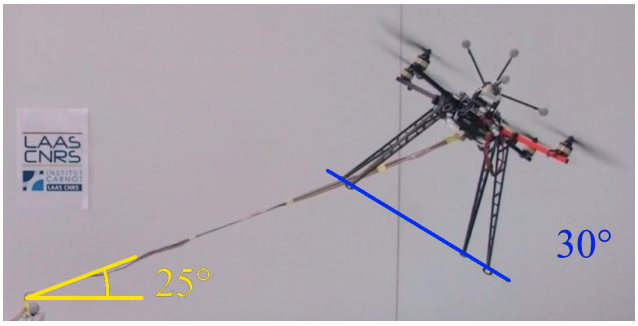


Figure 2: Inclined hovering with the robot tilted by 30° and with the cable elevation (represented in yellow) equal to 25°.

controlled in an decoupled way. In particular, thanks to the tether, the robot can hover in any point of the sphere defined by the cable constraint, with almost any orientations (not only horizontally as for the free-flight case). In particular the sphere is centered on the anchoring point and has the radius equal to the cable length. In Fig. 2 we show the robot hovering at a constant position and with an inclination of 30° with respect to the horizon.

In [6] we have shown that this property can be exploited to land the aerial robot in *inclined hovering* on any sloped surfaces. The capacity of performing the maneuver with practically zero velocity, angular velocity, acceleration, and angular acceleration makes the tethered method very robust, reliable, and safe. Thus much preferable over the free-flight method.

To control the system in [6] we designed a hierarchical controller based on the differential flatness. The controller can steer the tether configuration and the inclination of the robot along any desired trajectories. It is then used to track a spline-defined trajectory from the initial configuration to the desired landing configuration, performing the task (see Fig. 3). The take-off is done analogously.

3 Landing and Take-off on/from a Sloped Surface Tilted by 50 degrees

In this experiment we consider the plausible scenario where a quadrotor-like vehicle has to deploy a smaller robot or a sensor on a sloped surface tilted by 50°, shown in Fig. 3. The robot, equipped with a cable ending with a hook, starts from a non-tethered configuration on the ground. Therefore it has to anchor the other end of the cable to the surface to then perform the landing in a tethered configuration. Once the robot has landed on the desired spot and deployed the robot/sensor, it can take-off from the surface again exploiting the tether. Finally it can go back to the initial position after having detached the cable from the surface.

3.1 Anchoring Tools and Mechanisms

In order to pass from a free-flight configuration to a tethered one, a method to fix the end of the cable to the surface has to be found. The mechanism to do so strongly depends on the application scenario and in particular on the material of the slope. For example, in the previously mentioned scenario

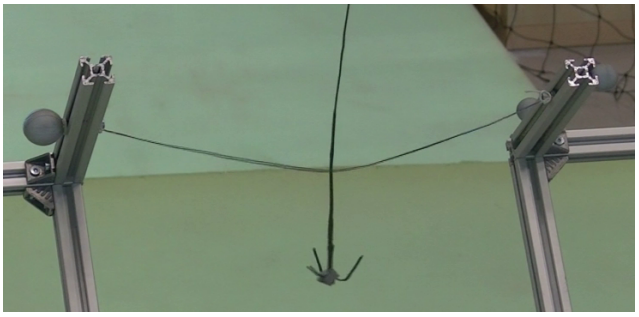


Figure 4: Zoom of the hook and the anchoring mechanism.

considered in the European project *Aeroarms*, the landing surface is mainly a pipe made of iron/steel. Thus in this case, and whenever the surface is made of proper metal, a magnetic anchor can be used to enhance the physical connection between surface and cable. In the case of a ground, snowed, or iced surface an harpoon-like mechanism might be envisaged.

In our experimental testbed we instead used a simpler solution based on a commercial fishing hook made of three tips, and an anchoring mechanism fixed to the surface made by an horizontal cable. In this way the robot can be tethered to the surface by sliding the vertical cable on the anchoring mechanism until the hook is anchored to the horizontal cable, as shown in Fig. 4. The hook can be detached from the anchoring mechanism doing the opposite operation. To facilitate this last action we increased the force pulling down the hook by slightly increasing its mass with an extra weight.

3.2 Experimental phases

Considering the previous experimental scenario and the goal, we divided the overall maneuver into several phases:

(a) approach to the anchor point with the hook,

(b) hooking of the anchoring system,

(c) stretching of the cable,

(d) tracking of the desired trajectory for tethered landing.

The phases from (a) to (c), described by the first row of images in Fig. 3, serve to pass from the initial free-flight configuration to the tethered one. Using a standard free-flight position controller and following a straight-line trajectory, the robot is able to anchor the anchoring system attached to the surface with the hook (see Fig. 3.b.2). The trajectory is planned such that the cable attached to the robot slides on the anchoring cable until the hook results attached to the last one.

Afterworld, during phase (c), the cable is stretched following a simple radial trajectory whose ending point is slightly outside the reachable region limited by the cable length. The robot, trying to reach this ending position, as explained in [9], will apply an extra force to the cable that will make it taut. In particular, the farther the desired ending position, the larger the internal force on the link. Using the dynamics of the system, the estimated state, and the control inputs, the robot can estimate the tension on the link. This estimation is then used to understand when the cable results sufficiently taut. Once the tension exceeds a certain security threshold a supervisor switches from the free-flight controller to the tethered one. Finally the planned landing trajectory is tracked. In order to compute the desired trajectories the parameters of the landing surface, such as slope angle and anchoring point, must be known. To acquire those numbers we applied some markers on the surface to measure its pose with a motion capture system. However, thanks to the robustness of the method, those parameters does not have to be very precise.

Once the robot ends the landing maneuver the take-off can start. The take-off is practically the play-back of the previous phases. Indeed, following the previous trajectory in the opposite sense lets the hook be de-attached from the anchoring mechanism to then go back to the starting point in a free-flight configuration.

3.3 Controller Switch

During the switching between the controllers, the continuity of the control input has to be guaranteed in order to preserve the stability of the system and to avoid undesired vibrations and jerks on the cable. This is obtained by setting as desired output of the next controller, the value of the system output at the switching instant. This is possible because for a specific output, there exist a unique input and state to obtain it. Therefore, asking to the next controller to remain in current state, it will requires the same input, thus preserving its continuity and the continuity of the full state.

3.4 Software Architecture

A schematic representation of the software architecture is represented in Fig. 5. The overall controllers and observers run on a ground PC. The desired spinning velocities of each propeller are sent at 500 [Hz] to the robot using a serial cable. The received velocity commands are then actuated by a controller (presented in [10]) running on the on-board ESC (Electronic Speed Control). The same serial communication is used to read at 1 [KHz] the IMU measurements that are then UKF-fused together with the motion capture system measurements (position and orientation of the quadrotor at 120 [Hz]) to obtain an estimation of the pose of the vehicle. The state estimation is then used to close the control loop and to get an estimation of the internal force along the link when the latter is taut.

The controller for the free-flight and tethered cases run in parallel and a supervisor, according to the state of the experiment, decides whose input has to be applied to the real system. The user input in the supervisor is needed to trigger situations of emergency.

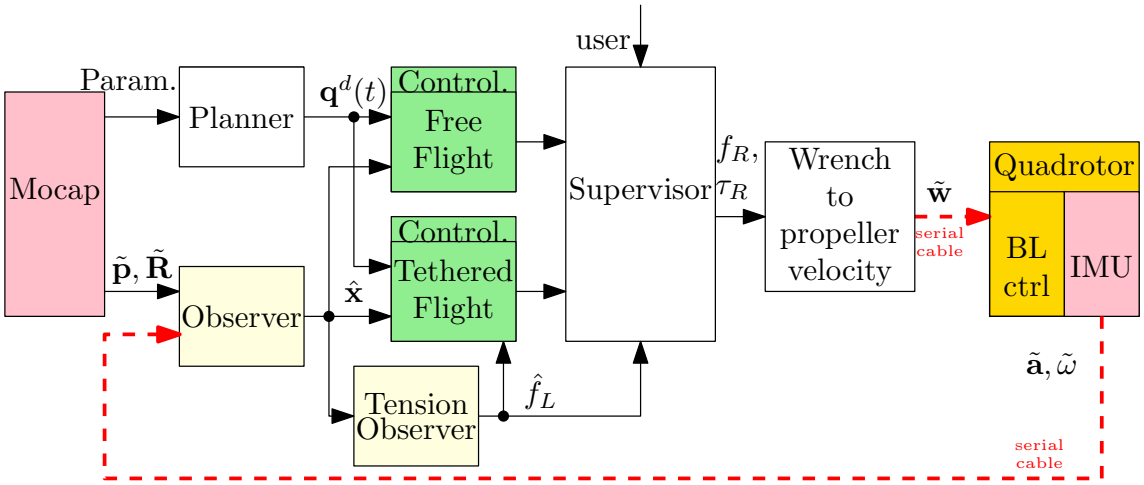


Figure 5: Schematic representation of the software architecture. Pink blocks represent the sensors. Green blocks represent the controllers and light yellow blocks represent the observers. Starting from the left, $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{R}}$ represent the measured robot position and orientation, respectively; $\hat{\mathbf{x}}$ and $\hat{f}_L$ represent the estimated state and link internal force, respectively; $\mathbf{q}^d(t)$ represents the desired output trajectory; f_R and τ_R represents the input of the robot, i.e., thrust intensity and torque vector; $\tilde{\mathbf{w}}$ represents the desired spinning velocity of the propellers. Finally $\tilde{\mathbf{a}}$ and $\tilde{\omega}$ represent the readings of the IMU, i.e., specific acceleration and angular velocity.

3.5 Nonideality

Another practical aspect that has to be considered is the non-zero offset between the cable attaching point and the vehicle center of mass. Indeed the controller presented in [6] assumes that this offset is equal to zero. In this way the robot translational and rotational dynamics can be decoupled. However, this never happens in a practical case. Then, due to this non-zero offset, the internal force along the cable generates a torque on the vehicle that has to be carefully compensated. This is done computing the extra torque from the estimated tension and the estimated offset calculated with a mechanical analysis.

Finally we highlight the fact that the maximum tiling of the surface is bounded by the input limits. Indeed the more inclined is the slope, the less it is the thrust required to compensate the gravity close to the surface. Due to the impossibility of producing negative thrust for the single propeller, the almost zero total thrust implies a reduced control authority on the total input moment that may cause the instability of the attitude dynamics and of the whole system in general.

3.6 Experimental Results

In Fig. 3 and 6 the experimental results are shown. Figure 3 shows the first half of the experiment, i.e., the landing, by a series of images. In particular the first row shows the anchoring procedure done in a free-flight condition. On the other hand, the second row shows the actual execution of the tethered landing. A video of the full experiment is available at `https://www.youtube.com/watch?v=01UYN289YXk&t=7s`

Figure 6 shows the evolution of the state, outputs and inputs of the system during the landing and take-off maneuvers. At time zero the tethered controller is activated and the landing maneuver starts. At time t_L the landing is accomplished and the surface is reached. At time t_G the motors are stop to simulate the deploying of a robot/sensor. Finally, at time t_T the take-off maneuver starts.

From those plots one can see that the desired trajectory is tracked precisely, with only some small errors due to calibration errors. Despite the presence of tracking errors the landing and take-off maneuver are accomplished successfully and in a very safe and gentle way. This shows the big advantage of using a tether that makes the execution on the task reliable and robust to tracking and modeling errors.

Furthermore, notice that the intensity of the internal force along the link, defined by the symbol f_L, is always positive. This shows that the cable is kept taut for the whole execution of the maneuvers.

4 Conclusion

Despite its practical issues, the use of the tether and the presented control method, greatly increases the robustness and the reliability of the maneuver with respect to the free-flight method. However for the real application some improvement of the system has to be considered. For example a small winch could be added to unroll and roll-up the cable immediately before and after the tethered maneuvers. A more suitable anchoring mechanism can be designed according to the type of landing surface. Finally the robot could be equipped with a on-board vision system to identify the posi-

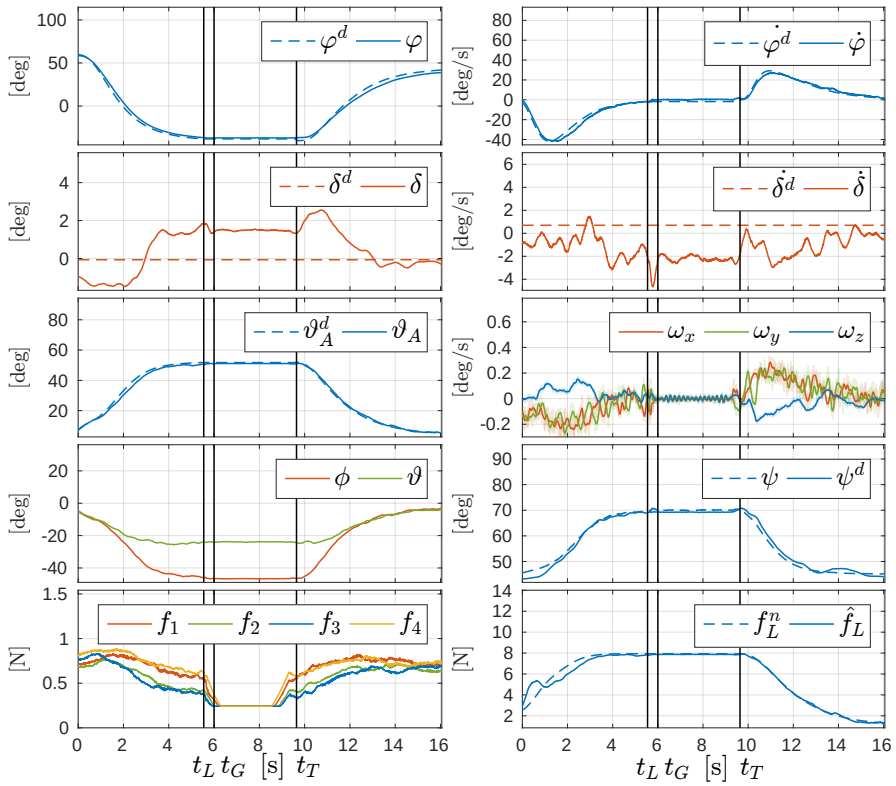


Figure 6: Plots of the state, outputs and inputs of the system during the tethered landing and take-off. In particular φ and δ describe the attitude on the cable and, given the link constraint, the position of the vehicle with respect to the anchoring point. ϑ_A is the angle between the robot and the link. ϕ and ψ are the angles that together with ϑ_A describe the orientation of the robot. f_1, f_2, f_3, f_4 are the forces produced by each propeller. Finally, f_L is the intensity of the internal force along the link. The super-script d and n represent the desired and the nominal values of the variable, respectively.

tion of anchoring point and the slope of the surface.

Acknowledgements

This research has been funded by the European Union's Horizon 2020 research and innovation program under grant agreement No 644271 AEROARMS.

References

[1] D. Mellinger, N. Michael, and V. Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The Int. Journal of Robotics Research*, 31(5):664–674, 2012.

[2] M. Tognon and A. Franchi. Dynamics, control, and estimation for aerial robots tethered by cables or bars. *IEEE Trans. on Robotics*, 33(4):834–845, 2017.

[3] M. M. Nicotra, R. Naldi, and E. Garone. Nonlinear control of a tethered uav: The taut cable case. *Automatica*, 78:174 – 184, 2017.

[4] S. Lupashin and R. D'Andrea. Stabilization of a flying vehicle on a taut tether using inertial sensing. In *2013 IEEE/RSJ Int. Conf. on Intell. Robots and Systems*, pages 2432–2438, Tokyo, Japan, Nov 2013.

[5] M. Tognon, S. S. Dash, and A. Franchi. Observer-based control of position and tension for an aerial robot tethered to a moving platform. *IEEE Robotics and Autom. Letters*, 1(2):732–737, 2016.

[6] M. Tognon, A. Testa, E. Rossi, and A. Franchi. Take-off and landing on slopes via inclined hovering with a tethered aerial robot. In *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1702–1707, Daejeon, South Korea, Oct. 2016.

[7] M. Tognon and A. Franchi. Position tracking control for an aerial robot passively tethered to an independently moving platform. In *20th IFAC World Congress*, Toulouse, France, Jul. 2017.

[8] L.A Sandino, D. Santamaria, M. Bejar, A. Viguria, K. Kondak, and A. Ollero. Tether-guided landing of unmanned helicopters without GPS sensors. In *2014 IEEE*

Int. Conf. on Robotics and Automation, pages 3096–3101, Hong Kong, China, May 2014.

[9] G. Gioioso, M. Ryll, D. Prattichizzo, H. H. Bülthoff, and A. Franchi. Turning a near-hovering controlled quadrotor into a 3D force effector. In *2014 IEEE Int. Conf. on Robotics and Automation*, pages 6278–6284, Hong Kong, China, May. 2014.

[10] A. Franchi and A. Mallet. Adaptive closed-loop speed control of BLDC motors with applications to multi-rotor aerial vehicles. In *2017 IEEE Int. Conf. on Robotics and Automation*, Singapore, May 2017.

Flight Simulation of a MAKO UAV for Use in Data-Driven Fault Diagnosis

Elgiz Baskaya,* Murat Bronz, and Daniel Delahaye
ENGIE Ineo - Groupe ADP - SAFRAN RPAS Chair, ENAC,
UAS Lab, ENAC,
Laboratoire MAIAA, ENAC

ABSTRACT

Last decade witnessed the rapid increase in number of drones of various purposes. This pushes the regulators to rush for safe integration strategies in a way to properly share the utilization of airspace. Accommodating faults and failures is one of the key issues since they constitute the bigger chunk in the occurrence reports available. The hardware limitations for these small vehicles point the utilization of analytical redundancy rather than the usual practice of hardware redundancy in the conventional flights. In the course of this study, fault detection and diagnosis for aircraft is reviewed. Then a nonlinear model for MAKO aircraft is simulated to generate faulty and nominal flight data. This platform enables to generate data for various flight conditions and design machine learning implementations for fault detection and diagnosis.

1 INTRODUCTION

Unmanned Aircraft Systems (UAS) are becoming more efficient platforms everyday for scientific/commercial domains offering benefits in terms of cost, flexibility, endurance as well as realizing missions that would be impossible with a human onboard. Increasing usage of these vehicles for a variety of missions, such as defense, civilian tasks including transportation, communication, agriculture, disaster mitigation applications pushes demand on the airspace. Furthermore, this congestion is predicted to accelerate with the growing diversity of these vehicles[1].

Improvement of the reliability of the flight is considered to be one of the main goals for integrating UAVs into civil airspace according to Unmanned systems roadmap by US Office of the Secretary of Defense, DoD [2]. To achieve a safe flight is not an easy task considering the unknowns of the systems hardware, environment and possible system faults and failures to emerge. Also, increasing demand on cost effective systems, resulting in the smaller sensors and actuators with less accuracy, impose the software to achieve even more. The expectation that UAVs should be less expensive than their manned counterparts might have a hit on reliability of the system. Cost saving measures other than the need to support a pilot/crew onboard or decrement in size would probably lead to decrease in system reliability.

Systems are often susceptible to faults of different nature. Existing irregularities in sensors, actuators, or controller could be amplified due to the control system design and lead to failures. A fault could be hidden thanks to the control action [3].

Under the research and development programs and initiatives identified by DoD in order to develop technologies and capabilities for UAS, the biggest chunk in control technologies is the health management and adaptive control with a budget of 74.3 M dollars. Other safety features such as validation and verification of flight critical intelligent software is the second with 57.8 M dollars [2].

The widely used method to increase reliability is to use more reliable components and/or hardware redundancy. Both requires an increase in the cost of the UAS conflicting one of the main reasons of UAS design itself band consumer expectations [4]. To offer solutions for all different foreseen categories of airspace, a variety of approaches should be considered. While hardware redundancy could cope with the failure situations of UAVs in the certified airspace, it may not be suitable for UAVs in open or some subsets of specific categories due to budget constraints. Analytical redundancy is another solution, may be not as effective and simple as hardware redundancy, but relies on the design of intelligent methods to utilize every bit of information onboard aircraft wisely to deal with the instances.

There are three approaches to achieve safe FTC in standard flight conventions. First one is the fail operational systems which are made insensitive to any single point component failure. The second approach is the fail safe systems where a controlled shut down to a safe state is practiced whenever a critical fault is pointed out by a sensor. The level of degradation assures to switch to robust (alternate) or direct (minimal level of stability augmentation independent of the nature of the fault) mode. Switching from nominal mode to the robust and direct modes leads to a decrease in the available GNC functions. This causes a degradation in ease of piloting. And also some optimality conditions could have been compromised. The third approach is fault tolerant control systems in which redundancy in the plant and the

*Email addresses: elgiz.baskaya@enac.fr, murat.bronz@enac.fr, daniel.delahaye@enac.fr

automation system is employed to design software that monitors the components and takes in action whenever needed. The strategy is most probably to try to keep plant availability and accept reduced performance [5].

RECONFIGURE project of FP7 [6] aims to attack at this problem of piloting degradation and optimality compromisation by attacking Flight Parameter Estimation (FPE) which is the online estimation of aircraft parameters, FDD and FTC in case of off-nominal events [7] They utilize a black box nonlinear model of aircraft and The project uses some outputs of a previous FP 7 project ADDSAFE leaded by Deimos Space [8].

2 Methods for FTCS

Since fault tolerant control is comprised of a set of different disciplines and a relatively new topic, the terminology is not solid. FDI could be a proper example to this ambiguity. In some works, it stands for Fault Detection and Isolation while in some other Fault Detection and Identification, which could also named after Fault Detection and Diagnosis, meaning that identification is added to Fault Detection and Isolation [9].

One of the first attempts to unify the terminology is carried out by IFAC SAFEPROCESS technical committee in 1996 and published by [10]. Fault, failure, and the methodology to handle those such as fault detection, fault isolation, fault identification, fault diagnosis and supervision terms explained separately to avoid the ongoing ambiguity in this field. Although fault detection methods are clearer in the work, difference between the methods for two steps of fault diagnosis, namely the fault isolation and fault identification is not very obvious.

Among different categorizations for the fault tolerance, there are options to handle faults on-line or off-line. Employing fault diagnosis schemes on-line is a way to achieve fault tolerance. In this case, as soon as a fault detected, a supervisory agent is informed via a discrete event signal. Then accommodation of the faults are handled either with the selection of a predetermined controller for the specific fault case, or by designing the action online with real-time analysis and optimization [5].

Another common categorization of FTCS is passive and active FTCS. In passive FTCS, the flight controller is designed in such a way to accommodate not only the disturbances but also the faults. Most of the times it a robust controller and does not require a diagnosis scheme. Active FTCS first distinguishes the fault via fault detection and diagnosis module and then switch between the designed controllers specific to the fault case or design a new one online [4]. While active FTCS requires more tools to handle faults as seen in Fig. 1, for faults not predicted and not counted for during the design of the robust controller, robust controller most probably fails.

Even with a long list of available methods, aerospace industry has not implemented FTC widely, except some space

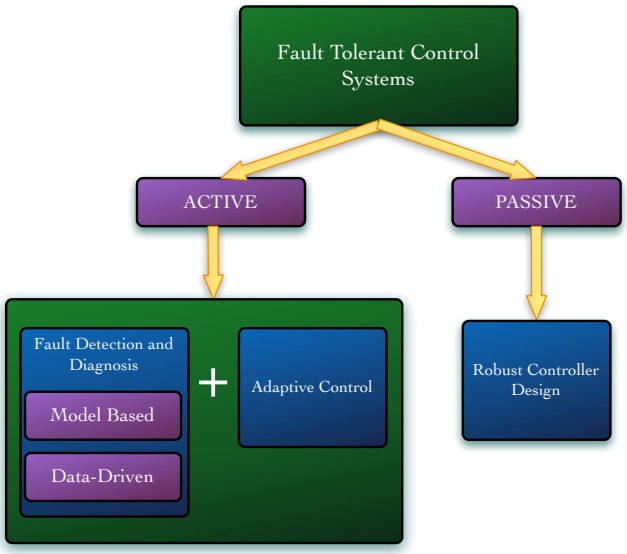


Figure 1: Variations of fault tolerant control systems

systems, due to the evolving nature of the methods, the tricks coming with the nonlinear nature of the problem, design complexity and high possibility of wrong alarms in case of large disturbances and/or modeling uncertainties. So the already carried reliability measures concerning the hardware redundancy is now the preferred way because of its ease and maturity being implemented on various critical missions with considering human lives.

3 Fault Detection and Diagnosis

FDD is handled in two main steps; fault detection and fault diagnosis. Fault diagnosis encapsulates fault isolation and fault identification. The methods for detection and diagnosis are investigated for their frequency of utilization separately for sensor, actuator, process and controller faults in [10]. FDD should not only be sensitive to the faults but also robust to the model uncertainties and external disturbances.

Two distinct options to proceed in analytical redundancy are the model based approaches and data-driven approaches. They form the two ends of a continuous solution set line, so utilizing them in a combination might end up with better solutions. Model based fault diagnosis highlights the components of a system and the connections in-between, and their corresponding fault modes. Data driven fault diagnosis rely on the observational data and prefers dense, redundant and with a frequency larger than the failure rate.

This work constitutes the basis for our research on fault detection. The idea to simulate the data using the MAKO model given here first, rather than utilizing flight data, is to start small in order to isolate some probable consequences

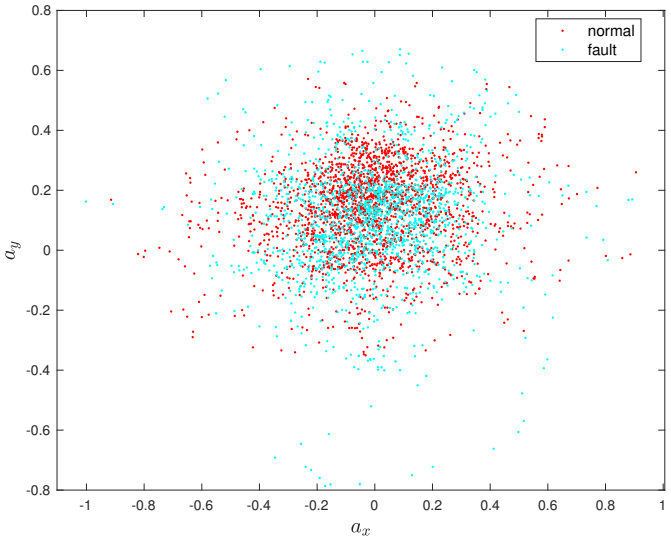


Figure 2: Accelerometer data readings a_x vs a_y



Figure 3: MAKO

such as the probable effect of the controller on the diagnosis. Detection of faults from real data is a challenging goal to start with as can be seen in Figure 2 real data accelerometer readings showing that it seems impossible to classify in case of a fault with one of the control surfaces is 50% less efficient.

Most of the FDI algorithms are implemented to open-loop systems, ignoring the probable influences of the controller might cause on the detection performance [11]. Here the system is open-loop as well. So we follow a step by step approach and hope to end with a more realistic case in the future, in which real flight data is utilized and diagnosis is achieved on-line aside a functioning controller. Here we present a literature survey for FTC of drones followed by efforts to deliver a full drone simulation which will serve as an environment to simulate measurements. A MAKO simulation is given in Matlab script which is freely available though the GIT [1] platform, using specifications MAKO, stability derivatives, aerodynamic force derivatives generated by AVL.

4 Methodology and Simulations

In this study, first, a model of an aircraft is simulated. This model, will not be used for the design of FDI algorithms, but instead will be utilized to test them. Nonlinear aircraft flight dynamics for translational and attitude motion can be given as a system of first order differential equations

$$\dot{\boldsymbol{x}}_n = \boldsymbol{C}_b^n \boldsymbol{v}^b \tag{1}$$

$$\dot{\boldsymbol{v}}^b = \frac{1}{m}\left[m\boldsymbol{g}^b + \boldsymbol{F}_t^b + \boldsymbol{F}_a^b\right] - {\boldsymbol{\omega}_{b/n}^b}^{\times} \boldsymbol{v}^b \tag{2}$$

$$\dot{q}_0 = -\frac{1}{2}\boldsymbol{q}_\nu^T \boldsymbol{\omega}_{b/n}^b \tag{3}$$

$$\dot{\boldsymbol{q}}_\nu = \frac{1}{2}\left(\boldsymbol{q}_\nu^{\times} + q_0 \boldsymbol{I}_3\right)\boldsymbol{\omega}_{b/n}^b \tag{4}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}}_{b/n}^b = \boldsymbol{M} - {\boldsymbol{\omega}_{b/n}^b}^{\times} \boldsymbol{J}\boldsymbol{\omega}_{b/n}^b \tag{5}$$

where $\boldsymbol{x}_n \in \mathbb{R}^3$ is the position of the center of mass of UAV in navigation frame $\mathcal{N}$, $\boldsymbol{v}^b$ is the velocity of the center of mass of UAV in body frame $\mathcal{B}$, $\boldsymbol{q} = [q_0, \boldsymbol{q}_v^T]^T \in \mathbb{R}^3 \times \mathbb{R}$ is the unit quaternion representing the attitude of the body frame $\mathcal{B}$ with respect to navigation frame $\mathcal{N}$ expressed in the body frame $\mathcal{B}$, $\boldsymbol{\omega}_{b/n}^b$ is the angular velocity of the body frame $\mathcal{B}$ with respect to navigation frame $\mathcal{N}$ expressed in the body frame $\mathcal{B}$, $\boldsymbol{J} \in \mathbb{R}^{3\times3}$ is the positive definite inertia matrix of the drone, $\boldsymbol{M} \in \mathbb{R}^3$ represents the moments acting on the drone, $\boldsymbol{C}_b^n$ is the direction cosine matrix which transforms a vector expressed in the body frame to its equal expressed in the navigation frame, $\boldsymbol{I}_3 \in \mathbb{R}^{3\times3}$ is the identity matrix, $\boldsymbol{F}_t^b \in \mathbb{R}^3$ is the thrust force expressed in the body frame, $\boldsymbol{F}_a^b \in \mathbb{R}^3$ are the aerodynamic forces given in the body frame. The navigation frame is assumed to be a local inertial frame in which Newton's Laws apply. The notation $\boldsymbol{x}^{\times}$ for a vector $x = [x_1 \quad x_2 \quad x_3]^{\mathrm{T}}$ represents the skew-symmetric matrix

$$\boldsymbol{x}^{\times} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{6}$$

[1] https://github.com/benelgiz/curedRone/tree/MAKOmodel

The stability derivatives and aerodynamic force coefficients are generated by AVL and given in Appendix A. AVL is an open source program developed at MIT and uses vortex-lattice method for the aerodynamic and stability calculations. The output of the program is linearized at a selected condition, therefore all the coefficients are calculated around the equilibrium point at $14m/s$ cruise flight condition. The center of gravity is located at $X_{CG} = 0.295\,m$, which corresponds to a $8\,\%$ of positive static margin that has been flight tested.

As an addition to the aerodynamic coefficients and stability derivatives, it is useful to have the moments of inertia of the aircraft so that one can use the model in a simulator. For that purpose, the aircraft is hanged by two strings, at different orientations, as shown in Figure 4, and measurements performed by timing the oscillation period for each axis. The resultant moment of inertias are given Table 2.

Further, the equations for calculation of forces and moments are given in Appendix B to simulate translational and rotational motion of a MAKO UAV.

The input vector can be written as $\boldsymbol{u}\,(t) \in \mathbb{R}^3$

$$\boldsymbol{u}\,(t) = \left[\delta_a\ \delta_e\ n\right]^{\mathrm{T}} \tag{7}$$

Here δ_a aileron deflection angle in degrees, δ_e elevator deflection angle in degrees, n engine speed in rev/s.

To validate the written translational and attitude motion dynamics and kinematics, MATLAB Simulink *6DOF* block has been utilized. This block accepts inputs as the force and moment and outputs the states of aircraft motion Fig. 6. To compare the generated model and Simulink *6DOF* block, forces and moments have been calculated via equations and constants given in Appendix A and Appendix B. The simulated states from the model script have been saved in advance and called from Simulink by *From Workspace* blocks then compared with the *6DOF* outputs. The difference found to be negligible indicating the validity of the model.

When the actuators are healthy, actual control input signal will be equal to the given input signal. In case of a fault the actual signal can be modeled as

$$\boldsymbol{u}\,(t) = \boldsymbol{E}\boldsymbol{u}_c + \boldsymbol{u}_f \tag{8}$$

where $\boldsymbol{u}_c$ is the desired control signal, $\boldsymbol{E} = diag(e_1, e_2, e_3)$ is the effectiveness of the actuators where $0 \leq e_i \leq 1$ with $(i = 1, 2, 3)$ and $\boldsymbol{u_f}$ additive actuator fault. This model makes it possible to simulate all four types of actuator faults shown in Fig. 5.

The measurements are simulated using the statistics of the hardware in the house. The sensor suit simulated is the InvenSense MPU-9250 Nine-axis (Gyro + Accelerometer + Compass) MEMS MotionTracking Device.

$$\boldsymbol{z}_{gyro} = \boldsymbol{k}_{gyro}\boldsymbol{\omega}^b_{b/i} + \boldsymbol{\beta}_{gyro} + \boldsymbol{\eta}_{gyro} \tag{9}$$

$$\boldsymbol{z}_{acc} = \boldsymbol{k}_{acc}\boldsymbol{\omega}^b_{b/i} + \boldsymbol{\beta}_{acc} + \boldsymbol{\eta}_{acc} \tag{10}$$

Here $\boldsymbol{\beta}$ is the bias, and $\boldsymbol{\eta}$ is the zero mean Gaussian process with $\boldsymbol{\sigma}^2$ variance and given in Table 3. For faulty and normal measurement values, drone model simulation which outputs the measurements as well should be run twice with different control surface input values. As an example, a faulty situation can be that even the controller gives an desired output of 4 degrees to the control surface, the control surface might have stuck at 1 degrees. Generated set of measurements can be visualized in feature space one by one. Such an example is the normalized accelerometer measurement components plotted Fig. 7.

It is always important to visualize the features to have a grasp of data structure. For that reason, available observations forms the 6-dimensional pattern space, $\boldsymbol{z} \in \mathbb{R}^6$ can be visualized in pairs to observe. There are further methods to visualize multidimensional data such as Tours methods [12, 13, 14], and GGobi data visualization system [15].

In this study, dimensionality reduction technique called Principle Component Analysis (PCA) is used for visualization. In PCA, the idea in general is to map the feature vector, $\boldsymbol{x} \in \mathbb{R}^n$ to a lower dimensional space where the new feature set will be represented by $\boldsymbol{z} \in \mathbb{R}^k$. Fig. 8 shows the resulted most significant elements for a mapped feature space from six dimensional feature vector to two. The structure of the data gives insight for the selection of some parameters or kernels for the purpose of classification. Here, it seems that a linear kernel is satisfactory by discarding the outliers. Another point is that the classifier might need a nice tuning due the the presence of outliers. The learning phase utilizes data including outliers and preciseness to fit the model to each of the data might end up an overfitted model, resulting in worse performance to generalize to new data coming in the prediction phase.

5 Conclusion

In this work, first a review on fault tolerant control for UAVs is given by pointing out its importance on today's challenging task of safe integration of drones into airspace. Data-driven methods for fault diagnosis is aimed to avoid the burden of modeling each craft especially considering for small drones it is not very realistic for most of the applications to have an accurate model for a variety reasons such as cost. AVL program is used to generate the coefficients for MAKO and a full simulation is realized. Statistics of the sensor suite in house is used for simulation of accelerometer and gyro data. For a preliminary investigation on data, six dimensional feature space is mapped to two dimensions via PCA for visualization purposes. The data shows that a linear kernel might be satisfactory for the purpose of two class classification. Due

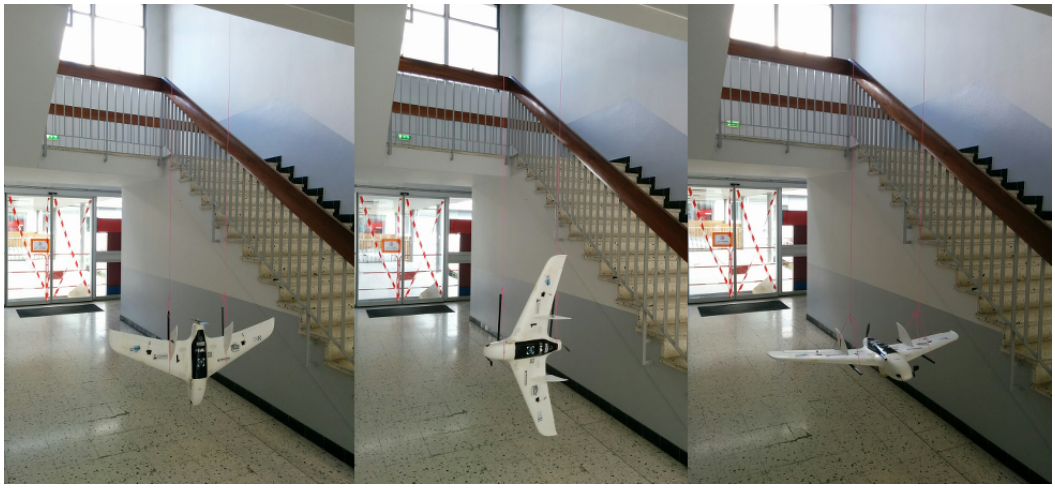


Figure 4: Moments of inertia measurements for each axis, I_{xx}, I_{yy}, I_{zz}.

to the presence of outliers, fine tuning or using optimization techniques could be needed to avoid overfitting or under fitting during the learning phase of the classification problem.

Acknowledgements

This work was supported by the ENGIE Ineo - Groupe ADP - SAFRAN RPAS Chair. Special thanks to Gautier Hattenberger and Torbjoern Cunis for code modifications to Paparazzi autopilot system, Xavier Paris, Michel Gorraz and Hector Garcia de Marina, and the rest of the ENAC Drone Lab for their help during test flights. Last but not least, we acknowledge Paparazzi community for their contributions to the autopilot system.

References

[1] Elgiz Baskaya, Guido Manfredi, Murat Bronz, and Daniel Delahaye. Flexible open architecture for uass integration into the airspace: Paparazzi autopilot system. In *Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th*, pages 1–7. IEEE, 2016.

[2] Unmanned Systems Roadmap 2005 - 2030, 2005.

[3] Guillaume JJ Ducard. *Fault-tolerant flight control and guidance systems: Practical methods for small unmanned aerial vehicles*. Springer Science & Business Media, 2009.

[4] Plamen Angelov. *Sense and avoid in UAS: research and applications*. John Wiley & Sons, 2012.

[5] Mogens Blanke, Christian W Frei, Franta Kraus, Ron J Patton, and Marcel Staroswiecki. What is fault-tolerant control. In *Preprints of 4th IFAC Symposium on Fault Detection Supervision ans Safety for Technical Processes, SAFEPROCESS*, pages 40–51, 2000.

[6] Philippe Goupil, Josep Boada-Bauxell, Andres Marcos, Paulo Rosa, Murray Kerr, and Laurent Dalbies. An overview of the fp7 reconfigure project: industrial, scientific and technological objectives. *IFAC-PapersOnLine*, 48(21):976–981, 2015.

[7] RECONFIGURE FP7 Project. reconfigure.deimos-space.com/, Accessed: 2016-07-19.

[8] ADDSAFE FP7 Project. http://addsafe.deimos-space.com/, Accessed: 2016-07-09.

[9] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, 32(2):229–252, 2008.

[10] Rolf Isermann and Peter Ballé. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control engineering practice*, 5(5):709–719, 1997.

[11] Rohit Pandita, József Bokor, and Gary Balas. Closed-loop performance metrics for fault detection and isolation filter and controller interaction. *International Journal of Robust and Nonlinear Control*, 23(4):419–438, 2013.

[12] Daniel Asimov. The grand tour: a tool for viewing multidimensional data. *SIAM journal on scientific and statistical computing*, 6(1):128–143, 1985.

[13] Dianne Cook and Andreas Buja. Manual controls for high-dimensional data projections. *Journal of computational and Graphical Statistics*, 6(4):464–480, 1997.

[14] Dianne Cook, Andreas Buja, Javier Cabrera, and Catherine Hurley. Grand tour and projection pursuit. *Journal of Computational and Graphical Statistics*, 4(3):155–172, 1995.

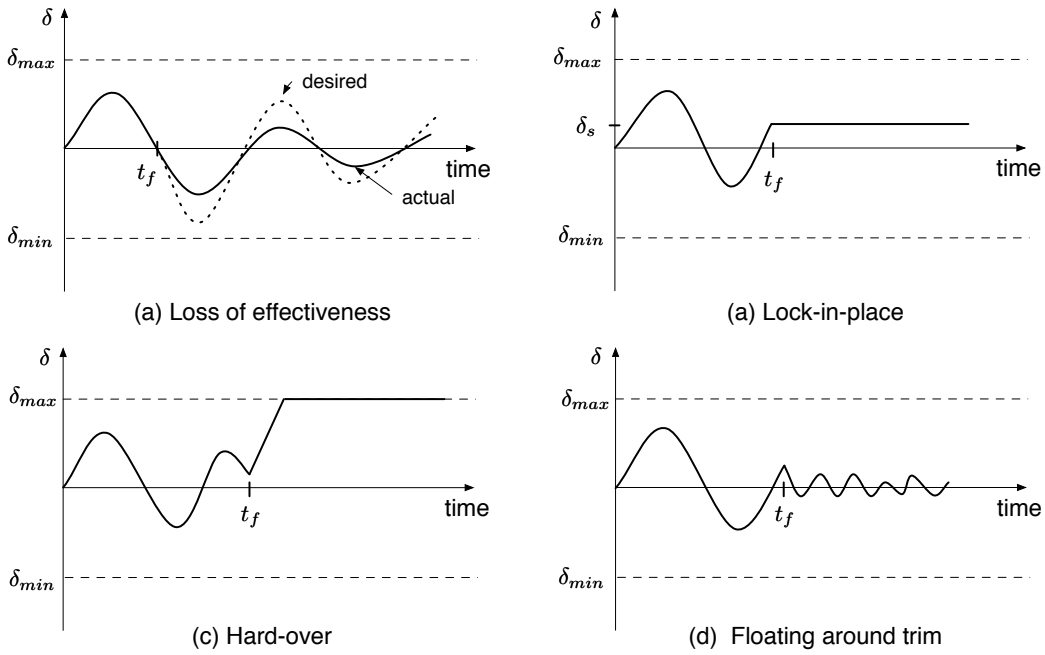


Figure 5: Common actuator faults [3]

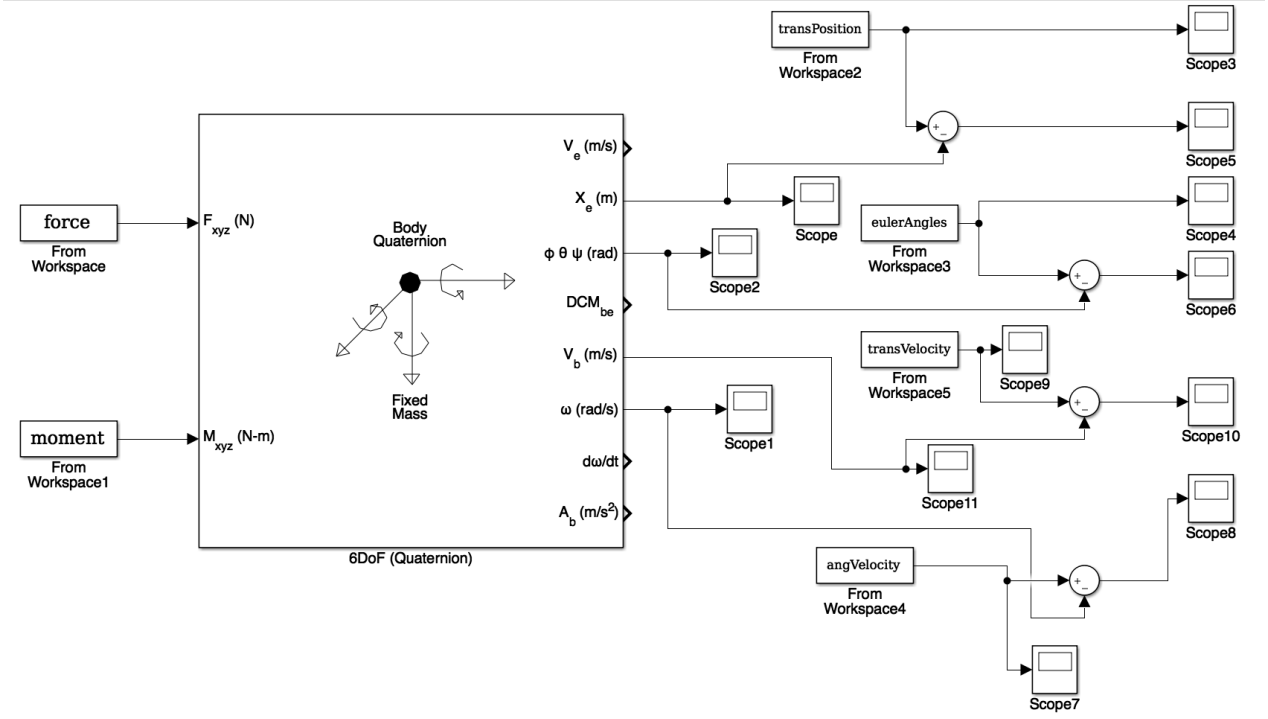


Figure 6: Validation with Simulink 6DOF aircraft model

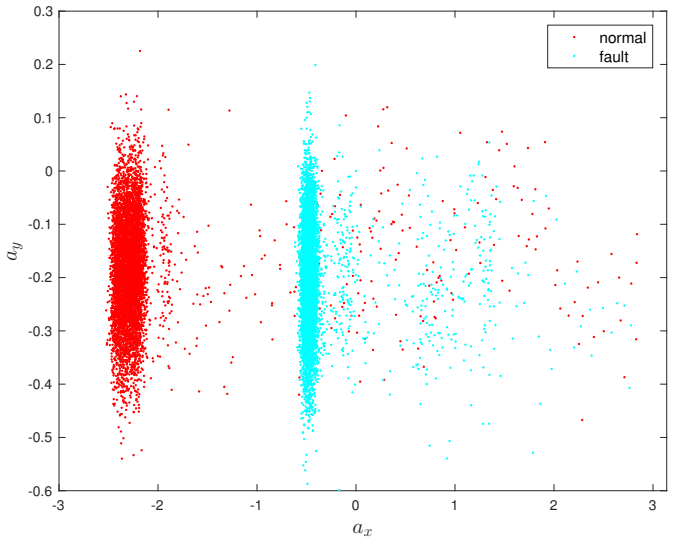


Figure 7: Accelerometer simulation a_x vs a_y

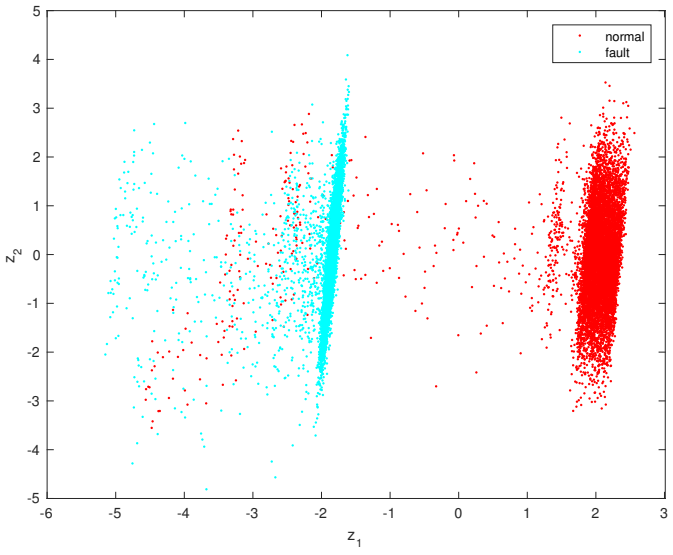


Figure 8: Reduced feature space z_1 vs z_2

[15] Dianne Cook and Deborah F Swayne. *Interactive and dynamic graphics for data analysis: with R and GGobi*. Springer Science & Business Media, 2007.

[16] Murat Bronz and Gautier Hattenberger. Aerodynamic characterization of an off-the-shelf aircraft via flight test and numerical simulation. In *AIAA Flight Testing Conference*, page 3979, 2016.

[17] Murat Bronz, Hector Garcia de Marina, and Gautier Hattenberger. In-flight thrust measurement using on-board force sensor. In *AIAA Atmospheric Flight Mechanics Conference*, page 0698, 2017.

[18] Jean-Philippe Condomines. *Développement d?un estimateur d?état non linéaire embarqué pour le pilotage-guidage robuste d?un micro-drone en milieu complexe*. PhD thesis, INSTITUT SUPERIEUR DE L'AERONAUTIQUE ET DE L'ESPACE (ISAE), 2015.

APPENDIX A: MAKO COEFFICIENTS

Table 1: General specifications of MAKO [16]

Parameter	**Value**	**Definition**
Wing span	1.288	$[m]$
Wing surface area	0.27	$[m^2]$
Mean aero chord	0.21	$[m]$
Take-off mass	$0.7 - 2.0$	$[kg]$
Flight velocity	$10 - 25$	$[m/s]$
I_{xx}	0.02471284	$[kg \cdot m^2]$
I_{yy}	0.015835159	$[kg \cdot m^2]$
I_{zz}	0.037424499	$[kg \cdot m^2]$

Table 2: Specifications of the sensor suit InvenSense MPU-9250 Nine-axis (Gyro + Accelerometer + Compass) MEMS MotionTracking Device[18]

Measurement	β	σ
z_{accx}	0.142	0.0319
z_{accy}	-0.3	0.0985
z_{accz}	0.19	0.049
z_{gyro_x}	-1.55	0.0825
z_{gyro_y}	-1.13	0.1673
z_{gyro_z}	-1.7	0.2214

Table 3: Stability derivatives for MAKO extracted from AVL program at $14m/s$ equilibrium cruise speed

Parameter	Value	Definition
C_{L_a}	-0.1956×10^{-2}	roll derivative
$C_{L_{\tilde{p}}}$	-4.095×10^{-1}	roll derivative
$C_{L_{\tilde{r}}}$	6.203×10^{-2}	roll derivative
C_{L_β}	3.319×10^{-2}	roll derivative
C_{M_0}	0	pitch derivative
C_{M_e}	-0.076×10^{-1}	pitch derivative
$C_{M_{\tilde{q}}}$	-1.6834	pitch derivative
C_{M_α}	-32.34×10^{-2}	pitch derivative
C_{N_a}	-0.0126×10^{-2}	yaw derivative
$C_{N_{\tilde{p}}}$	-4.139×10^{-2}	yaw derivative
$C_{N_{\tilde{r}}}$	-0.1002×10^{-1}	yaw derivative
C_{N_β}	2.28×10^{-2}	yaw derivative

Table 4: Aerodynamic force derivatives for MAKO extracted from AVL program at $14m/s$ equilibrium cruise speed

Parameter	Value	Definition
C_{Z_0}	-8.53×10^{-2}	lift derivative
C_{Z_α}	3.9444	lift derivative
C_{Z_q}	4.8198	lift derivative
C_{Z_e}	1.6558×10^{-2}	lift derivative
C_{X_0}	2.313×10^{-2}	drag derivative
C_{X_k}	1.897×10^{-1}	drag derivative
C_{Y_β}	-2.708×10^{-1}	side force derivative
$C_{Y_{\tilde{p}}}$	1.695×10^{-2}	side force derivative
$C_{Y_{\tilde{r}}}$	5.003×10^{-2}	side force derivative
C_{Y_a}	0.0254×10^{-2}	side force derivative

Table 5: Thrust force coefficients for propeller APC SF 9×6 from wind tunnel experiments [17]

Parameter	Value	Definition
$C_{F_{T1}}$	1.342×10^{-1}	thrust derivative
$C_{F_{T2}}$	-1.975×10^{-1}	thrust derivative
$C_{F_{T_{rpm}}}$	7.048×10^{-6}	thrust derivative
D	$0.228\,m$	propeller diameter

APPENDIX B: FORCE, MOMENT CALCULATIONS

Roll torque

$$L_B = \bar{q}\, S\, b\, C_L \tag{11}$$

$$C_L = C_{L_a}\, \delta_a + C_{L_{\tilde{p}}}\, \tilde{p} + C_{L_{\tilde{r}}}\, \tilde{r} + C_{L_\beta}\, \beta \tag{12}$$

Pitch torque

$$M_B = \bar{q}\, S\, \bar{c}\, C_M \tag{13}$$

$$C_M = C_{M_e}\, \delta_e + C_{M_{\tilde{q}}}\, \tilde{q} + C_{M_\alpha}\, \alpha \tag{14}$$

Yaw torque

$$N_B = \bar{q}\, S\, b\, C_N \tag{15}$$

$$C_N = C_{N_a}\, \delta_a + C_{N_{\tilde{p}}}\, \tilde{p} + C_{N_{\tilde{r}}}\, \tilde{r} + C_{N_\beta}\, \beta \tag{16}$$

$$\bar{q} = \frac{\rho V_T^2}{2} \tag{17}$$

Lift force

$$Z^w = \bar{q}\, S\, C_Z(\alpha) \tag{18}$$

$$C_Z(\alpha) = C_{Z_0} + C_{Z_\alpha}\alpha \tag{19}$$

Drag force

$$X^w = \bar{q}\, S\, C_Z(\alpha, \beta) \tag{20}$$

$$C_X(\alpha) = C_{X_1} + C_{X_k}\, C_Z^2 = C_{X_1} + C_{X_k}\, (C_{Z_1} + C_{Z_\alpha}\alpha)^2 \tag{21}$$

Lateral force

$$Y^w = \bar{q}\, S\, C_Y(\beta) \tag{22}$$

$$C_Y(\beta,\, \tilde{p},\, \tilde{r},\, \delta_a) = C_{Y_\beta}\beta + C_{Y_{\tilde{p}}}\, \tilde{p} + C_{Y_{\tilde{r}}}\, \tilde{r} + C_{Y_a}\, \delta_a \tag{23}$$

Thrust force model

$$F_T = \rho n^2 D^4 C_{F_T} \tag{24}$$

$$C_{F_T} = C_{F_{T1}} + C_{F_{T2}} J + C_{F_{T3}} J^2 \tag{25}$$

$$J = \frac{V_T}{n\pi D} \tag{26}$$

Incremental Nonlinear Dynamic Inversion and Multihole Pressure Probes for Disturbance Rejection Control of Fixed-wing Micro Air Vehicles

Elisabeth S. van der Sman,* Ewoud J. J. Smeur, Bart Remes, Christophe De Wagter, and Qiping Chu
Delft University of Technology, 2629 HS Delft, The Netherlands

ABSTRACT

Maintaining stable flight during high turbulence intensities is challenging for fixed-wing micro air vehicles (MAV). Two methods are proposed to improve the disturbance rejection performance of the MAV: incremental nonlinear dynamic inversion (INDI) control and phase-advanced pitch probes. INDI uses the angular acceleration measurements to counteract disturbances. Multihole pressure probes measure the incoming flow angle and velocity ahead of the wing in order to react to gusts before an inertial response has occurred. The performance of INDI is compared to a traditional proportional integral derivative (PID) controller with and without the multihole pressure probes. The attitude controllers are tested by performing autonomous wind tunnel flights and stability augmented outdoor flights. This paper shows that INDI improves the disturbance rejection performance of fixed-wing MAVs compared to traditional proportional integral derivative controllers.

1 INTRODUCTION

THE number of Micro Air Vehicles (MAVs) flying in urban areas is increasing due to the low cost, the low weight, the availability of ready-to-use platforms and the variety of applications. Fixed-wing MAVs are ideal for tasks which require a long range and endurance such as mapping the environment, surveillance, photography and delivering goods. These tasks may require the MAV to fly between buildings and obstacles which generate high energy turbulence [1]. The turbulence intensity profile increases as the MAV flies closer to the ground reaching levels up to 50% [2]. The perceived turbulence level depends on the MAV flight speed. The lower the flight speed the higher the turbulence intensity, indicating that hover is the most critical condition for MAVs [3]. Fixed-wing MAVs are particularly susceptible to wind gusts due to the large wing area [3]. Mohamed et al. identified two main approaches to counteract turbulence: reactive and phase-advanced [2]. Reactive techniques use sensors to measure the inertial response to disturbances. This paper focuses on developing a reactive nonlinear controller in combination with phase-advanced sensors to increase the stability of fixed-wing MAVs flying through turbulence.

Gusts cause large variations in airspeed and attitude angles in MAV, leading to nonlinear behavior. To maintain an optimal performance gain scheduling is needed [4]. Alternatively, using the model based Nonlinear Dynamic Inversion (NDI) a uniform performance over the flight envelope is guaranteed if the aerodynamic model is accurately known [5]. The major disadvantage of NDI is the sensitivity to model mismatch [6]. Determining an accurate aerodynamic model of the MAV is very expensive and time consuming. It has been very successful in programs like the Lockheed Marting X-35 [7][8] and the NASA X-36 tailless aircraft [9]. Incremental Nonlinear Dynamic Inversion (INDI) [10][11][6] has been proposed to highly reduce model sensitivity. It relies on angular acceleration, filtered from differentiated angular rate provided by the gyroscopes[12]. Smeur et al. [12] found that the same filter should be used on the actuators to provide time synchronization with the measured acceleration. The only model parameters required by the INDI attitude controller are the actuator dynamics and the control effectiveness [12]. INDI has been implemented on fixed-wing MAVs for the first time by Vlaar [13].

Phase-advanced sensors are used to measure turbulence before an inertial response has occurred [14][15]. The first phenomenon to occur is the change in flow pitch angle and velocity ahead of the wing. These variations cause an uneven lift distribution over the wings leading to structural stresses which can be measured by strain sensors [16]. Mohamed et al. [2] developed a pitch probe sensor inspired by the leading edge feathers of birds. This sensor measures the variations in the angle of attack and speed of the incoming flow. By measuring the incoming gust ahead of the wing a time advantage is created. The gust is related to the pressure distribution over the wing which in turn causes an angular acceleration measurement. The sensor was therefore placed at the point of highest correlation between the surface pressure variation and the measured angular acceleration [16]. The gust measurement is used as a feed-forward component to each aileron separately to locally counteract the gust on each wing [2]. This sensor enhances the controller performance by decreas-

*Msc. Student, Control and Simulation Department, Faculty of Aerospace Engineering

ing the range of the roll and pitch angle displacements [2].

Wind tunnel flights were performed by inserting static turbulence generating grids at the inlet of the test section. The distance from the grids in the direction of the flow determines the turbulence intensity level experienced by the MAV [17]. Outdoor tests are performed with natural turbulence. The length scale of indoor turbulence is in the order of 1 meter compared to 15 meters for outdoor flights [2]. The aim of this paper is to investigate the disturbance rejection performance of INDI applied to fixed-wing MAVs in high turbulence intensities and to compare the performance of the proposed system with a traditional PID controller. Finally research is performed on the incorporation of the pitch probe sensors in the INDI control structure. All systems are implemented on the Slick 360 Micro fixed-wing MAV with the Open-Source Paparazzi Autopilot system. The disturbance rejection performance is tested in the same turbulence intensity conditions which are typical for urban environments.

The structure of this paper is the following: Section 2 describes the MAV model, Section 3 is dedicated to the design of the INDI and PID controllers and Section 4 describes the pitch probe sensors. Section 5 presents the experimental setup. The results are presented in Section 6.

2 MAV Model

The sum of moments experienced during flight by the Slick 360 Micro fixed-wing MAV expressed in the body frame are described by Eq. 1 [6] [18].

$$\dot{\mathbf{\Omega}} = \mathbf{I}^{-1}(\mathbf{M} - \mathbf{\Omega} \times \mathbf{I}\mathbf{\Omega}) \tag{1}$$

The sum of moments around the body axes $\{X_B, Y_B, Z_B\}$ is given by $\mathbf{M}^T = \begin{bmatrix} M_x & M_y & M_z \end{bmatrix}$. The angular rates are denoted by $\mathbf{\Omega}^T = \begin{bmatrix} p & q & r \end{bmatrix}$. The moment of inertia matrix is defined by Eq. 2 assuming a plane of symmetry around the longitudinal and vertical axis $(X_B Z_B - plane)$.

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix} \tag{2}$$

The resulting system of equations is given by Eq. 3 [18] [19].

$$\begin{aligned} M_x &= I_{xx}\dot{p} + (I_{zz} - I_{yy})qr - I_{xz}(\dot{r} + pq) \\ M_y &= I_{yy}\dot{q} + (I_{xx} - I_{zz})rp + I_{xz}(p^2 - r^2) \\ M_z &= I_{zz}\dot{r} + (I_{yy} - I_{xx})pq - I_{xz}(\dot{p} + rq) \end{aligned} \tag{3}$$

Expressed in terms of the angular accelerations Eq. 3 becomes Eq. 4 [19].

$$\begin{aligned} \dot{p} &= (c_1 r + c_2 p)q + c_3 M_x + c_4 M_z \\ \dot{q} &= c_5 pr - c_6(p^2 - r^2) + c_7 M_y \\ \dot{r} &= (c_8 p - c_2 r)q + c_4 M_x + c_9 M_z \end{aligned} \tag{4}$$

The definitions of the multiplication parameters c_1 up to c_9 with $\Gamma = I_{xx}I_{zz} - I_{xz}^2$ [19]:

$$\begin{aligned} &\Gamma c_1 = (I_{yy} - I_{zz})I_{zz} - I_{xz}^2 && \Gamma c_3 = I_{zz} \\ &\Gamma c_2 = (I_{xx} - I_{yy} + I_{zz})I_{xz} && \Gamma c_4 = I_{xz} \\ &c_5 = (I_{zz} - I_{xx})I_{yy}^{-1} && c_6 = I_{xz}I_{yy}^{-1} \\ &c_7 = I_{yy}^{-1} && c_8 = (I_{xx} - I_{yy})I_{xx} - {I_{xz}}^2 \\ &\Gamma c_9 = I_{xx} \end{aligned}$$

The moments can be spit into two components Eq. 5: a part depending on the aerodynamic state variables and a part influenced by the control surfaces of the vehicle.

$$\mathbf{M} = \mathbf{M_a} + \mathbf{M_c} \tag{5}$$

Substituting Eq. 5 into Eq. 3 leads to Eq. 6 around the roll axis.

$$\begin{aligned} \dot{p} = (c_1 r + c_2 p)q + c_3 M_{x_a}(u, v, w, p, q, r) + \\ c_3 M_{x_c}(V, \delta_a, \delta_e, \delta_r) + c_4 M_{z_a}(u, v, \dot{v}, w, p, q, r) + \\ c_4 M_{z_c}(V, \delta_a, \delta_e, \delta_r)) \end{aligned} \tag{6}$$

In the body frame the components of the free stream velocity V are defined as u, v, w. The control surface deflections are denoted by $\delta_a, \delta_e, \delta_r$ for the ailerons, elevator and rudder respectively. Around the pitch axis substituting Eq. 5 into Eq. 3 leads to Eq. 7.

$$\begin{aligned} \dot{q} = c_5 pr - c_6(p^2 - r^2) + c_7 M_{y_a}(u, v, w, \dot{w}, p, q, r) + \\ c_7 M_{y_c}(V, \delta_a, \delta_e, \delta_r, \delta_t) \end{aligned} \tag{7}$$

A second derivative term is incorporated for the vertical velocity component $\dot{w}$. The thrust of the propeller is defined as δ_t.

Euler angles are used to define the orientation of the body frame with respect to the earth frame [18]. The kinematic attitude equations Eq. 8 for a flat non-rotating earth are used to relate the angular rates to the Euler angles [18].

$$\begin{aligned} \dot{\phi} &= p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \\ \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\psi} &= q\tfrac{\sin\phi}{\cos\theta} + r\tfrac{\cos\phi}{\cos\theta} \end{aligned} \tag{8}$$

3 Incremental Nonlinear Dynamic Inversion

The angular accelerations around the body axes defined by Eq. 4 can be written in an incremental form by applying a Taylor series expansion. The resulting equation is used by the controller to predict the angular acceleration one step ahead in time based on the current time point [12].

3.1 Roll axis

Three eigenmotions characterize the behavior of the MAV around the roll axis: the aperiodic roll, the dutch roll and the spiral. The fast aperiodic roll is used to model the angular acceleration based on aileron inputs. During this maneuver the MAV is flown at a constant speed V of 10m/s with $\Delta u = 0$. The yawing motion and rudder input are neglected: $\Delta v = 0 \quad \Delta\dot{v} = 0 \quad \Delta r = 0 \quad \Delta\delta_r = 0$. Due

to this assumption the yaw moment M_z can be omitted [18]. The asymmetric and symmetric motions are considered decoupled. The effects of the symmetric motions are neglected: $\Delta\delta_e = 0 \quad \Delta w = 0 \quad \Delta q = 0$. Applying a Taylor series expansion to Eq. 6 with the above mentioned assumptions leads to Eq. 9.

$$\dot{p} = \dot{p}_0 + \frac{\partial(c_3 M_{x_a})}{\partial p}\bigg|_{p=p_0}(p - p_0) + \frac{\partial(c_3 M_{x_c})}{\partial \delta_a}\bigg|_{\delta_a=\delta_{a0}}(\delta_a - \delta_{a0}) \tag{9}$$

The partial derivative of the aerodynamic moment $\frac{\partial(c_3 M_{x_a})}{\partial p}$ is defined as F_p. The partial derivative of the control moment $\frac{\partial(c_3 M_{x_c})}{\partial \delta_a}$ is defined as G_{δ_a}, simplifying Eq. 9 to Eq. 10.

$$\Delta\dot{p} = F_p \Delta p + G_{\delta_a}\Delta\delta_a \tag{10}$$

A least-square fitting method is used to determine F_p and G_{δ_a}. Data was collected by performing outdoor test flights. In total 90 seconds of flight test data are used, 80% for the training set and 20% for the test set. An airspeed controller was used to maintain the cruise speed at 10m/s. During the flight p and δ_a were logged. $\dot{p}$ can be calculated by differentiating the angular rate. The change in $\dot{p}$ is too noisy to be used directly for the fit, all signals are therefore filtered with the same second order low pass filter given by Eq. 11 with $w_n = 15.9$ Hz and $\zeta = 0.65$.

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{11}$$

The root mean square error of the test set and training set differ by 2% indicating the model is not over-fitted. The model parameters are $F_p = -16 \pm 1\ [1/s]$ and $G_{\delta_a} = 212 \pm 6\ [rad/(s^2 rad)]$. The INDI controller is based on the principle of time scale separation [12] [6] simplifying Eq. 10 to Eq. 12.

$$\Delta\dot{p} = G_{\delta_a}\Delta\delta_a \tag{12}$$

The angular acceleration prediction of the INDI controller with damping Eq. 10 and without damping Eq. 12 are compared and shown in Figure 1. The root mean square error of the model without damping is 0.4% higher than with damping.

3.2 Pitch axis

Around the pitch axis the short period motion is modeled during cruise at 10m/s, with $\Delta u = 0$ [18]. The thrust input is kept constant: $\Delta\delta_t = 0$. The asymmetric motions are neglected leading to $\Delta v = 0 \quad \Delta p = 0 \quad \Delta r = 0 \quad \Delta\delta_a = 0 \quad \Delta\delta_r = 0$. The second order derivative is neglected $\Delta\dot{w} = 0$. With these assumptions Eq. 7 is simplified

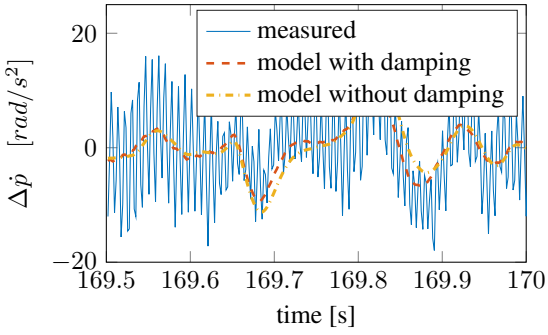


Figure 1: Measured and modeled angular acceleration: roll axis, $10m/s$

to Eq. 13.

$$\dot{q} = \dot{q}_0 + \frac{\partial(c_7 M_{y_a})}{\partial w}\bigg|_{w=w_0}(w - w_0) + \frac{\partial(c_7 M_{y_a})}{\partial q}\bigg|_{q=q_0}(q - q_0) + \frac{\partial(c_7 M_{y_c})}{\partial \delta_e}\bigg|_{\delta_e=\delta_{e0}}(\delta_e - \delta_{e0}) \tag{13}$$

The partial derivatives of the aerodynamic moment are defined as $F_q = \frac{\partial(c_7 M_{y_a})}{\partial q}$ and $F_w = \frac{\partial(c_7 M_{y_a})}{\partial w}$. The control effectiveness is defined as $G_{\delta_e} = \frac{\partial(c_7 M_{y_c})}{\partial \delta_e}$ leading to Eq. 14.

$$\Delta\dot{q} = F_w V \Delta\alpha + F_q \Delta q + G_{\delta_e}\Delta\delta_e \quad \alpha = w/V \tag{14}$$

To measure the angle of attack the MAV is placed in a pitch rig set-up where the MAV can only rotate around the pitch axis through the center of gravity. In this set-up the angle of attack α is considered equal to the pitch angle θ. A least-square fitting is used to determine F_w, F_q and G_{δ_e}. Open loop doublet inputs were applied in the wind tunnel at a speed of 10m/s. In total 45 seconds of flight test data are considered, 80% for the training set and 20% for the test set. During the flight q, θ and δ_e are logged. $\dot{q}$ is obtained by differentiating the angular rate. To decrease the noise level, all signals are filtered with the same second order low pass filter given by Eq. 11 with $w_n = 15.9$ Hz and $\zeta = 0.65$. The root mean square error of the test set and training set differ by only 0.2%. The model parameters are $F_w = -31.7 \pm 0.3\ [\frac{rad}{ms}]$, $F_q = -8.3 \pm 0.2\ [1/s]$ and $G_{\delta_e} = 73 \pm 1\ [rad/(s^2 rad)]$. The damping term F_w and the control effectiveness G_{δ_e} are in the same order of magnitude. This indicates the principle of time scale separation is theoretically no longer valid [6]. It is however, difficult to predict the value of the term $\Delta\alpha$ for the next time point. The INDI controller is therefore designed based on Eq. 15 considering errors can be present due to the effect of the damping term F_w.

$$\Delta\dot{q} = G_{\delta_e}\Delta\delta_e \tag{15}$$

The root mean square error of the model without damping is 22% higher compared to the model with damping. The increase in root mean square error is mainly due to the open loop response which includes a slow damping motion which is not captured by the initial elevator input. The damping is slower compared to the initial change in acceleration due to the elevator input and can therefore be compensated by a closed loop control structure. The comparison of the two models and the measured change in angular acceleration is given in Figure 2.

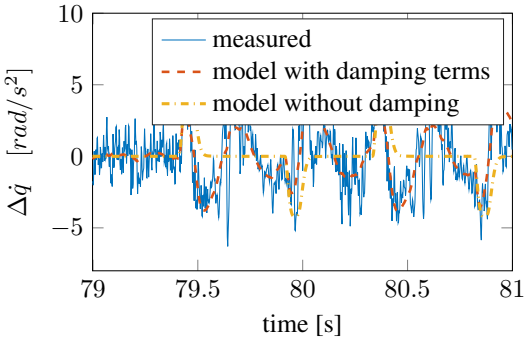


Figure 2: Measured and modeled angular acceleration: pitch axis, $10m/s$

3.3 Implementation

The INDI control scheme has the form given in Figure 3 and Figure 4 based on Eq. 12 and Eq. 15. The virtual control input v denotes the reference acceleration of the system which is compared to the measured acceleration $\dot{p}_f$ for roll and $\dot{q}_f$ for pitch. The subscript f is used to denote all signals which have been filtered with second order low pass filter $H(s)$. The inverse of the control effectiveness G is used to calculate the required change in input based on the angular acceleration error. This change in input cannot be achieved instantaneously but is filtered by the actuator dynamics $A(s)$. The actuator position that is achieved after a time step is fed back into the system delayed by the same filter $H(s)$ to achieve time synchronization with the angular acceleration. The linear PD controller is used to control the attitude angles. The controller uses the angular rate to calculate the derivative term. For small pitch angles this assumption is valid as shown by Eq. 8.

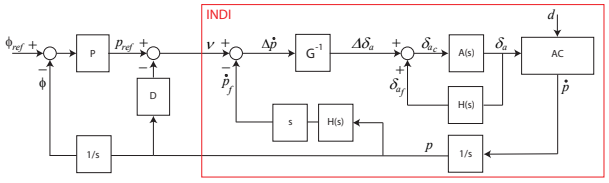


Figure 3: INDI control block structure roll

The actuators determine the reference tracking and disturbance rejection performance of the INDI controller [12].

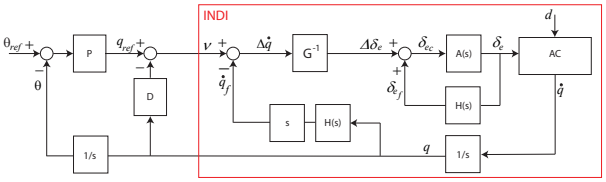


Figure 4: INDI control block structure pitch

The Slick 360 Micro uses 4 HK5330 micro servos to move the control surfaces. These servos are the fastest in this category currently available with a speed of 0.04 sec from 0 to 60 deg. The servo model is determined by logging the PWM command given by the autopilot and the position of the servo arm measured by the servo potentiometer. The servo is modeled as a first order system with a limited rate Eq. 16 and an initial delay of 10 milliseconds.

$$\begin{array}{l} A(s) = \frac{60}{s+60} \\ \left(\frac{\Delta\delta_a}{\Delta t} G_{\delta_a}\right)_{max} = 11 \quad [rad/s] \\ \left(\frac{\Delta\delta_e}{\Delta t} G_{\delta_e}\right)_{max} = 4 \quad [rad/s] \end{array} \tag{16}$$

3.4 Closed-loop Analysis

The closed-loop performance of the system can be calculated by simplifying the INDI controller to the actuator dynamics block shown in Figure 5 [12]. The actuator dynamics block contains a first order system and a rate limiter which is influenced by the control effectiveness G. Due to the rate limiter, increasing the PD gains does not necessarily lead to a faster reference tracking performance. During high turbulence intensities the system is considered to be operating in the range influenced by the rate limiter. The closed-loop response is analyzed for a step input of 0.2 rad for pitch and 0.4 rad for roll. Figure 6 shows the performance of the model with the parameters from Eq. 16 for $P = 100, 185, 300$ and 400. The ratio between the proportional and derivative gain should remain constant leading to a corresponding set of derivative gains $D = 12, 22, 36$ and 48. This shows that a drawback of using a higher P gain is the amplification of the noise in the roll and pitch rate signals due to a higher D gain. Figure 6 shows that the response of the system for a range of proportional gains is identical except for the last part of the step response where the first order system determines the behavior.

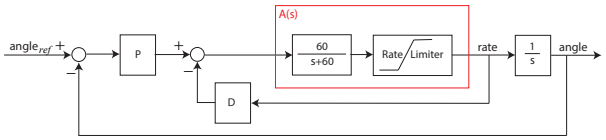
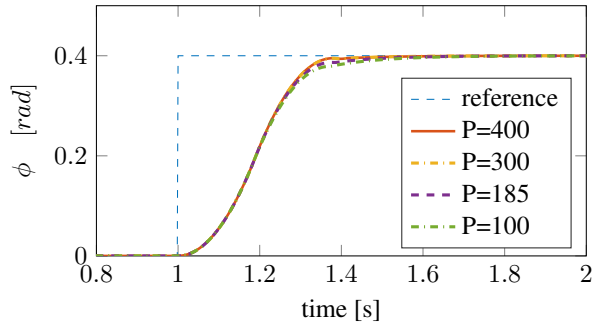
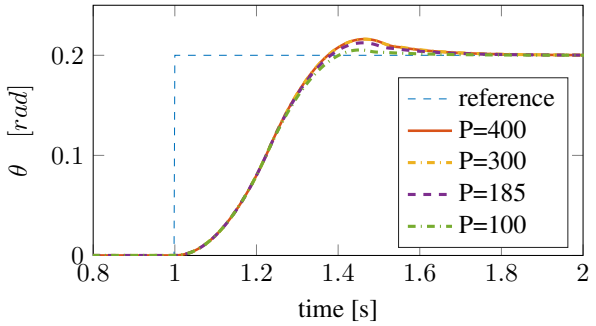


Figure 5: Linear gains analysis

Around the roll axis, the linear gains are designed to obtain a rise time of 0.22 sec given a step input of 0.4 rad without overshoot. This leads to $P = 185$ and $D = 22$. For pitch,



(a) Roll angle tracking of a step input of 0.4 radians



(b) Pitch angle tracking of a step input of 0.2 radians

Figure 6: Reference tracking performance for varying PD gains

the reference tracking for a step input of 0.2 rad is designed to give a rise time of 0.23 sec and a small overshoot of 6%. This leads to the same set of gains $P = 185$ and $D = 22$.

The PID gains used in this paper were manually tuned to obtain comparable rise time and overshoot properties as the INDI controller. For roll, the manually tuned P gain is 1.6 times higher compared to the INDI gain. The D gain used in the PID structure is 0.7 times lower compared to the INDI gain.

4 Phased-advanced pitch probes

The pressure probes system designed by Mohamed et al. [2] is used in the same configuration. The probe head is connected to a differential pressure sensor through acrylic tubes embedded in the wings. The probe is placed 15 cm ahead of the wing to create a 15 milliseconds time advantage at cruise speeds. The signals are feed-forwarded to both ailerons separately as shown in Figure 7 [2]. The probes sense changes in angle of attack of the incoming flow. Mohamed et al. [2] showed that a linear relationship between the angle of attack measurement and the differential pressure measurement. The probes are high pass filter using a fourth order Butterworth filter given by Eq. 17. Based on flight test data, the cut-off frequency was selected at 4Hz.

$$H(z) = \frac{b(1)+b(2)z^{-1}+b(3)z^{-2}+b(4)z^{-3}+b(5)z^{-4}}{a(1)+a(2)z^{-1}+a(3)z^{-2}+a(4)z^{-3}+a(5)z^{-4}} \quad (17)$$

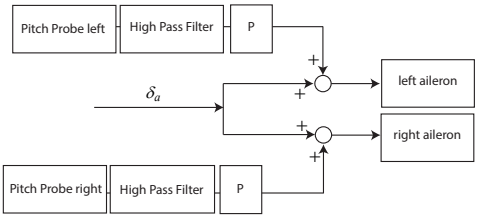


Figure 7: Feed-forward control probes [2]

with $b = \begin{bmatrix} 0.7194 & -2.8774 & 4.3162 & -2.8774 & 0.7194 \end{bmatrix}$ and $a = \begin{bmatrix} 1 & -3.3441 & 4.2389 & -2.4093 & 0.5175 \end{bmatrix}$.

5 Experimental set-up



Figure 8: Slick 360 Micro

The tests presented in this paper are all performed with the Slick 360 Micro shown in Figure 8. The MAV weighs 130 grams and has a wing span of 49 cm. The MAV is equipped with the LISA M Paparazzi open-source autopilot system and the phase-advanced pitch probes developed by Mohamed et al. [2]. Tests were performed in the Open Jet Facility (OJF) at Delft University of Technology and outdoors during a windy day. To ensure the human element is not influencing the results the MAV is flown completely autonomously with a vertical, longitudinal and lateral position control system.

5.1 Open Jet Facility

The OJF tunnel cross section is 285×285 cm. In the wind tunnel the coordinate system is defined as: X_w to the right perpendicular to the flow direction, Y_w in the direction of the tunnel flow and Z_w to the top of the wind tunnel. The origin of the axis system is given in Figure 9 in the middle of the cross section underneath the tunnel inlet. The head-

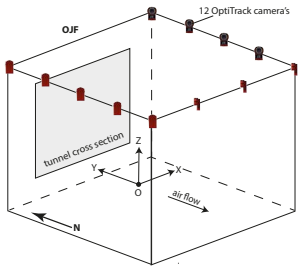


Figure 9: OJF wind tunnel coordinate system definition

ing is defined as 0 deg in the flow direction. The position is measured by 12 Optitrack cameras and sent at 20Hz.

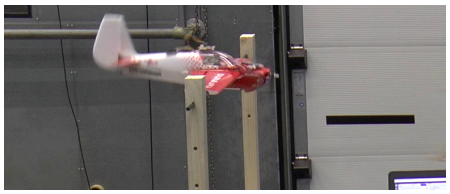


Figure 10: Pitch rig set-up

The pitch rig wind tunnel tests were performed to analyze the short period motion. The rig consists of a wooden construction attached to the tunnel floor and a movable rod connected to the wooden frame through low friction bearings. The rotational axis of the rig passed through the quarter chord line which corresponds to the center of gravity of the MAV. The pitch rig set-up is shown in Figure 10.

To replicate high levels of turbulence intensity static grids were placed at the test section inlet. The grids are built up of metal rods evenly spaced in the vertical direction and connected at the edges as shown in Figure 11. Close to the grids wake turbulence is generated which slowly decays into homogeneous turbulence [17]. This decay is captured by the variation of the turbulence intensity in the flow direction. The results shown in Section 6 are therefore always compared at the same $\{X_w, Y_w, Z_w\}$ position.



Figure 11: Turbulence grids set-up

5.2 Position Control

The MAV has to fly in a box with a maximum cross section of 285 $\times$ 285 cm minus the wing span. The position in the flow direction has to be constant to ensure the same level of turbulence intensity is experienced throughout multiple flights. In the wind tunnel the course angle cannot be used as the position of the MAV is constant and the MAV is effectively hovering. A heading based controller is used instead to control the lateral position.

5.2.1 Vertical and longitudinal control

The throttle is used to control the longitudinal displacements. During outdoor flights the throttle is only used for the climb and descent phases and therefore always combined with a corresponding pitch angle. The vertical controller uses the altitude error to calculate the reference climb rate. The altitude error is the difference between the reference altitude defined in the flight plan and the altitude z measured by the OptiTrack system. Based on climb rate, a feed-forward increase in throttle level on top of the cruise throttle is commanded. The longitudinal position y is compared to the reference position. A standard PID control structure is used to keep the aircraft at one location in the tunnel, as shown in Figure 12.

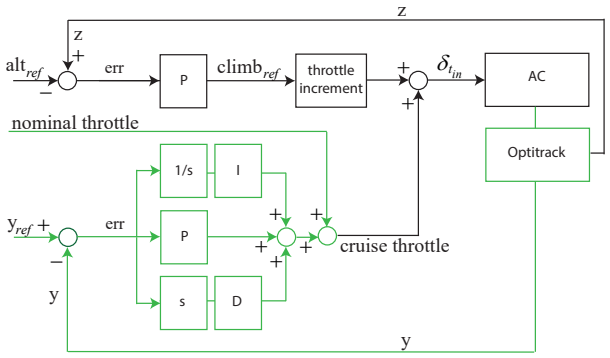


Figure 12: Vertical and longitudinal throttle control

The reference pitch angle is calculated from the desired climb rate which, as shown in Figure 13, depends on the altitude error.

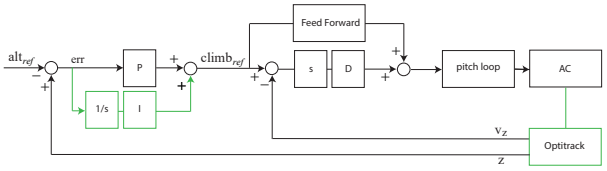


Figure 13: Vertical pitch control

5.2.2 Lateral control

The lateral controller defines a reference course (heading) angle χ_{ref} as the arctangent of the x and y distances between the MAV and the desired waypoint. A PD controller calculates the roll angle setpoint based on the heading error. An overview of the lateral position control block is given in Figure 14.

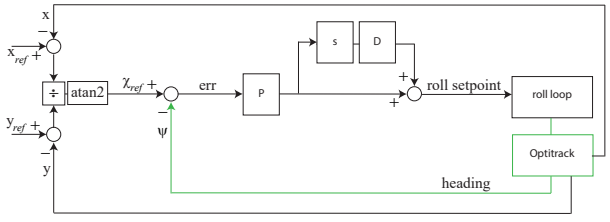


Figure 14: Lateral roll control

5.3 Outdoor test flights

The flight tests were performed in a field surrounded by trees to ensure the MAV is constantly subject to turbulence. The speed of the MAV was kept constant at 10m/s by a proportional integral (PI) airspeed controller regulating the throttle. The attitude was controlled by the test pilot. The turbulence length scale can be estimated by using the von Karman model given by Eq. 18 [17].

$$S_{\bar{u}\bar{u}}(f) = \frac{4\sigma^2 L_x}{V} \frac{1}{\left(1 + 70.8\left(\frac{L_x f}{V}\right)^2\right)^{\frac{5}{6}}} \tag{18}$$

$S_{\bar{u}\bar{u}}$ is the power spectral density as a function of the frequency f expressed in Hz. The power spectral density follows a -5/6 decay law as can be seen by the denominator of the function. σ is the standard deviation of the flow and L_x the turbulence length scale.

6 Results

Table 1: Preliminary and final parameters of the INDI controllers

	Roll		Pitch	
	Preliminary	Final	Preliminary	Final
P	400	185	400	185
D	22	22	22	22
G	0.019	0.022	0.014	0.008
ω_n	3.2	15.9	1.6	15.9

Table 2: Preliminary and final parameters of the PID controllers

	Roll		Pitch	
	Preliminary	Final	Preliminary	Final
P	12500	15000	4500	16000
I	1	30	1	30
D	700	700	1.5	1.5

6.1 Wind Tunnel Flights

Approximate values of the control effectiveness, linear gains and filter cut-off frequency were used during the initial OJF windtunnel tests. The parameters calculated in Section 3 are summarized in Table 1 and Table 2 and compared to the empirically determined preliminary parameters used during the autonomous wind tunnel tests.

The INDI controller proved to be very precise and able to maintain the position within a 1x1x1 meter box during all flights. The performance of INDI and PID is evaluated for the same time frame and for the the same position $\{X_w, Y_w, Z_w\}$ and is shown in Figure 17.

The results given in Figure 15 and Figure 16 show that the range of the probability density function for the roll angle error decreases by 40% with the INDI controller compared to the PID controller. For the pitch angle the range

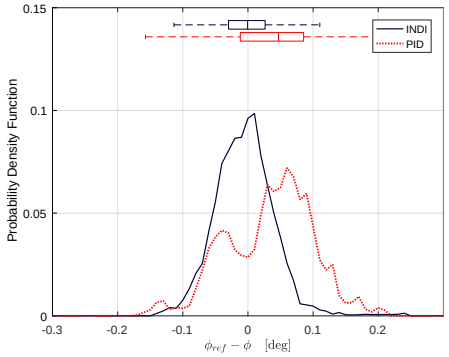


Figure 15: Roll angle perturbation for PID and INDI controller

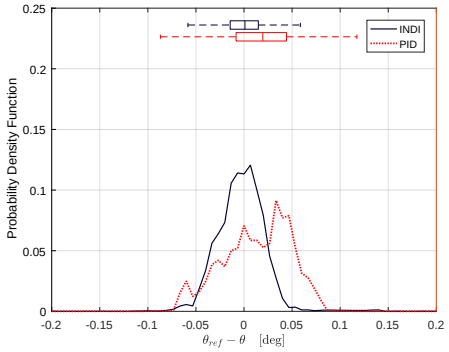


Figure 16: Pitch angle perturbation for PID and INDI controller

decreases by 43% for INDI compared to PID. The probability density functions of the PID controller for roll and pitch are not centered around zero. This indicates that the integral gains should be increased to obtain a better performance. The integral gains were therefore increased by a factor 30 for the final test flights, as shown in Table 2.

6.2 Outdoor flights

The reference tracking performance of the INDI controller around the roll axis is shown in Figure 18(a). The rise time of the step input given from 0 to 0.4 radians is on average 0.15 sec which is faster than the expected rise time obtained with the closed loop analysis in subsection 3.4 of Section 3. Overshoot is an average value of 10%. The reference tracking performance of the PID controller around the roll axis is shown in Figure 18(b) with an average rise time of 0.19 sec and overshoot of 5%. These tests were performed consecutively outdoors on the same day.

The reference tracking performance of the INDI controller around the pitch axis is shown in Figure 19(a). The rise time for the test performed at 10.6 m/s is 0.37 sec and the overshoot 13%. The variation between the three curves is due to the high levels of turbulence experienced during the flight and the different airspeeds at which the step inputs were given. The PID step response test was executed twice during this flight. The performance is difficult to assess due to the

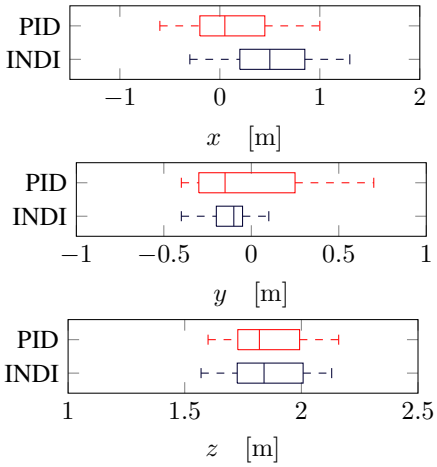
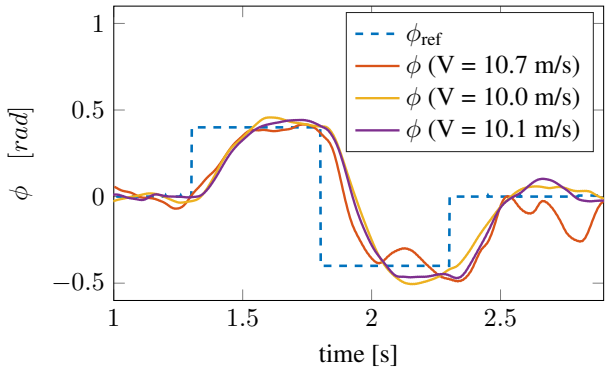
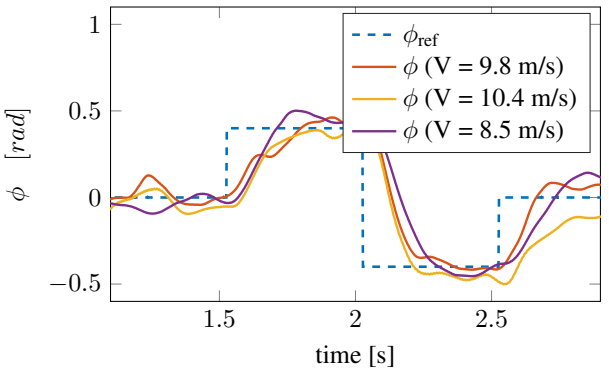


Figure 17: Position variation comparison between INDI and PID



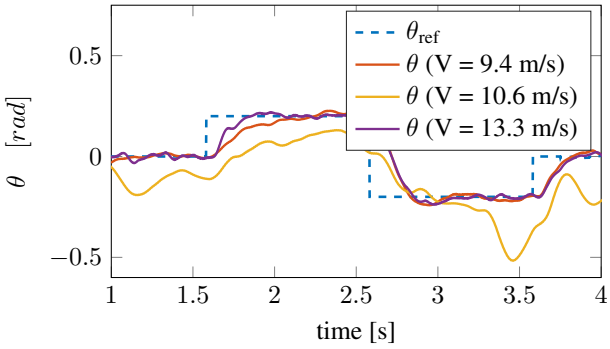
(a) INDI



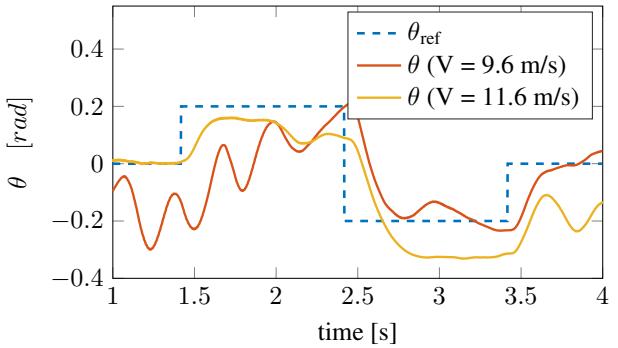
(b) PID

Figure 18: Roll axis tracking performance.

influence of high energy turbulence as shown in Figure 19(b). On a calm day the results obtained with the same PID gains show a very high performance as given in Figure 20. A small steady state-error is visible in the results indicating the tuning of the integral gain can be improved to eliminate the steady-state offset.



(a) INDI



(b) PID

Figure 19: Pitch axis tracking performance.

The disturbance rejection of the INDI controller and the PID controller with and without pitch probes was tested on the same day while flying the same trajectory multiple times. The analysis of all test flights showed that one on the pitch probes got obstructed during landing causing a bias in the measurements of the subsequent flights. Therefore no reliable data was obtained on the performance of the INDI controller in combination with the pitch probe sensors. The disturbance rejection performance is therefore analyzed for the other three control approaches. In total 120sec of reliable flight data are analyzed in this section, 40 seconds for each controller. The turbulence intensity level of the flight data is $T_i = 12.9\%$. The turbulence length scale is estimated by using the von Karman spectrum Eq. 18 to be 2.5 meters. The performance is evaluated for the part of the flight which used the airspeed controller to maintain the average velocity around 10m/s. The average velocity of the flight test data is 9.7 m/s.

Figure 21(a) clearly shows an improvement in the probability density function of the roll angle error for the INDI

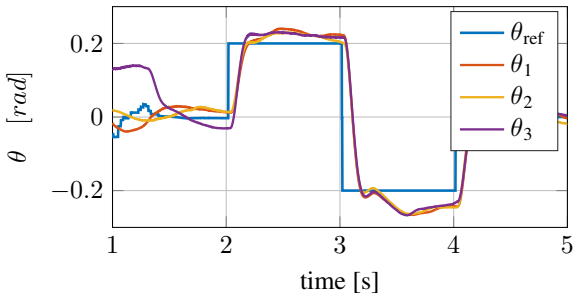


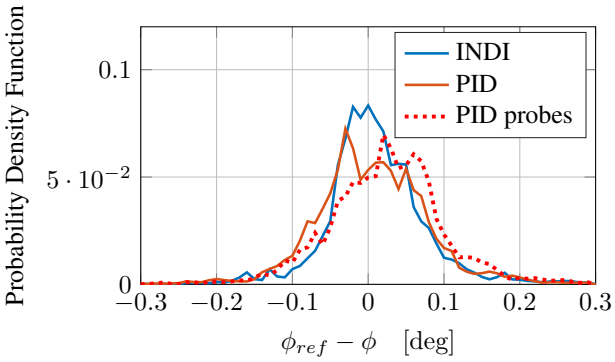
Figure 20: Pitch angle during doublet (PID, calm air, $10m/s$)

controller compared to the PID controller. The box plot of the data given in Figure 21(b) shows that the range decreases by 21% for INDI compared to PID. The enhanced PID controller with the pitch probes does not show an improvement in performance compared to the traditional PID controller. This result is not as expected from literature. Two main reasons have been identified which can influence the results: the high pass filter and degraded servos. The flight data show that the filtered probe values contain offsets. These offsets should have been eliminated by the high pass filter. The parameters used for the high pass filter during testing had a precision of 10^{-4}. By increasing the accuracy of the filter parameters to 10^{-6} all offsets are removed from the data. Another factor which was not taken into account is the degrading performance of the servos due to overheating. To eliminate this aspect the servos should be replaced after each test flight. Additional flight tests should be performed with accurate filter parameters and new servos to improve the performance of the pitch probe sensors.

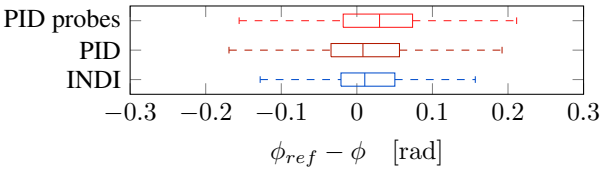
The pitch angle error shown in Figure 22(a) also shows a clear improvement with INDI compared to PID. The box plot highlights the difference in Figure 22(b) which indicates that the range decreases by 24% for INDI compared to PID.

7 Conclusion

This research shows the potential of incremental nonlinear dynamic inversion applied to fixed-wing micro air vehicles flown in high turbulence intensities. The angular acceleration measurements are predicted based upon the control surface deflections eliminating the need for a complex aerodynamic model. To test the performance of the system a novel control solution is presented which allows autonomous free flight in a wind tunnel. The throttle is used for both the longitudinal an vertical control and the heading angle is used for lateral control. This experimental set-up eliminates all human factors and provides the opportunity to test in turbulence intensities which are beyond human capabilities. Autonomous free flight wind tunnel tests in turbulence were performed with estimates of the control effectiveness and the filter cut-off frequency. The performance of the estimated model is presented as it illustrates the robustness of the system. The

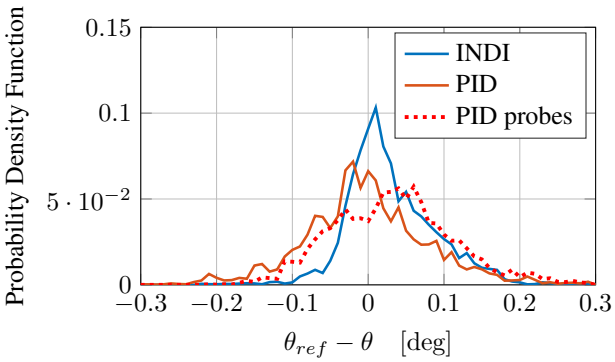


(a) Probability density function of the roll angle error

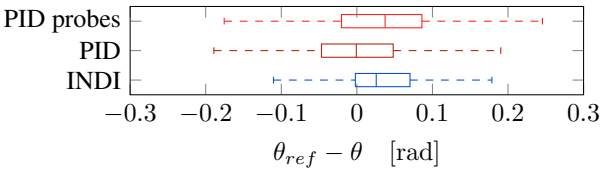


(b) Box plot of the roll angle error

Figure 21: Disturbance rejection performance around the roll axis



(a) Probability density function of the pitch angle error



(b) Box plot of the pitch angle error

Figure 22: Disturbance rejection performance around the pitch axis

results show that the controller eliminates all steady-state errors and counteracts accelerations caused by external disturbances making it a suitable solution for precise movements in the wind tunnel. Accurate model parameters for the control effectiveness, the actuators and the filter cut-off frequency are determined in this paper and used to assess the performance of the system during outdoor test flights. During outdoor flights the performance is compared to a proportional integral derivative controller tuned to obtain the same reference tracking performance. The nonlinear incremental controller significantly improves the disturbance rejection performance around both the roll and pitch axis. More research should be performed to assess the performance of the nonlinear controller compared to an enhanced linear controller with phase-advanced pitch probes.

REFERENCES

[1] S. Watkins, A. Mohamed, A. Fisher, R. Clothier, R. Carrese, and D. F. Fletcher. Towards autonomous mav soaring in cities: Cfd simulation, efd measurement and flight trials. *International Micro Air Vehicles Conference and Flight Competition 2015*, 2015.

[2] A. Mohamed, M. Abdulrahim, S. Watkins, and R. Clothier. Development and flight testing of a turbulence mitigation system for micro air vehicles. *Journal of Field Robotics*, 33(5):639–660, 2016.

[3] S. Watkins, J. Millbank, B. J. Loxton, and W. H. Melbourne. Atmospheric winds and their implications for microair vehicles. *AIAA Journal*, 44(11):2591–2600, 2006.

[4] J. J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall, 1991.

[5] H. P. Lee, S. E. Reiman, C. H. Dillon, and H. Youssef. Robust nonlinear dynamic inversion control for a hypersonic cruise vehicle. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2007.

[6] S. Sieberling, Q. Chu, and A. J. Mulder. Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. *AIAA Journal of Guidance, Control, and Dynamics*, 33(6):1732–1742, 2010.

[7] K. Bordignon and J. Bessolo. Control allocation for the x-35b. *2002 Biennial International Powered Lift Conference and Exhibit*, 2002.

[8] G. P. Walker and D. A. Allen. X-35b stovl flight control law design and flying qualities. *2002 Biennial International Powered Lift Conference and Exhibit*, 2002.

[9] J. S. Brinker and K. A. Wise. Flight testing of reconfigurable control law on the x-36 tailless aircraft. *AIAA Journal of Guidance, Control, and Dynamics*, 24(5):903–909, 2001.

[10] P. R. Smith. A simpified approach to nonlinear dynamic inversion based flight control. *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 1998.

[11] B. J. Bacon and A. J. Ostroff. Reconfigurable flight control using nonlinear dynamic inversion with a special accelerometer implementation. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2000.

[12] E. J. J. Smeur, Q. Chu, and G. C. H. E. de Croon. Adaptive incremental nonlinear dynamic inversion for attitude control of micro aerial vehicles. *AIAA Journal of Guidance, Control, and Dynamics*, 39(3):450–461, 2016.

[13] M. C. Vlaar. Incremental nonlinear dynamic inversion flight control implementation and flight test on a fixed wing uav. Master thesis, Delft University of Technology, Delft The Netherlands, February 2014.

[14] A Mohamed, R Clothier, S Watkins, R Sabatini, and M Abdulrahim. Fixed-wing mav attitude stability in atmospheric turbulence, part 1: Suitability of conventional sensors. *Progress in Aerospace Sciences*, 70:69–82, 2014.

[15] A Mohamed, S Watkins, R Clothier, M Abdulrahim, K Massey, and R Sabatini. Fixed-wing mav attitude stability in atmospheric turbulencepart 2: Investigating biologically-inspired sensors. *Progress in Aerospace Sciences*, 71:1–13, 2014.

[16] A. Mohamed, S. Watkins, R. Clothier, and M. Abdulrahim. Influence of turbulence on mav roll perturbations. *International Journal of Micro Air Vehicles*, 6(3):175–190, 2014.

[17] Sridhar Ravi. *The influence of turbulence on a flat plate aerofoil at Reynolds numbers relevant to MAVs*. PhD thesis, RMIT, Melbourne, Victoria, Australia, 2011.

[18] M. V. Cook. *Flight Dynamics Principles*. Elsevier, 2007.

[19] Lars Sonneveldt. *Adaptive Backstepping Flight Control for Modern Flighter Aircraft*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2010.

Human-Robot Cooperation in Surface Inspection Aerial Missions

Martin Molina[1], Pedro Frau[1], Dario Maravall[1]
Jose Luis Sanchez-Lopez[2,3], Hriday Bavle[2], Pascual Campoy[2]
[1]Department of Artificial Intelligence, Technical University of Madrid (UPM), Spain
[2]Centre for Automation and Robotics, UPM-CSIC, Spain
[3]Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg

ABSTRACT

The goal of the work presented in this paper is to facilitate the cooperation between human operators and aerial robots to perform surface inspection missions. Our approach is based on a model of human collaborative control with a mixed initiative interaction. In the paper, we present our human-robot cooperation model based on the combination of a supervisory mode and an assistance mode with a set of interaction patterns. We developed a software system implementing this interaction model and carried out several real flight experiments that proved that this approach can be used in aerial robotics for surface inspection missions (e.g., in vision based indoor missions). Compared to a conventional tele-operated inspection system, the solution presented in this paper gives more autonomy to the aerial systems, reducing the cognitive load of the operator during the mission development.

1 INTRODUCTION

Certain types of missions in aerial robotics may require special human-robot interaction with intermediate degrees of robot autonomy between manual teleoperation and complete autonomy. One example of this type of mission is surface inspection in which the operator uses aerial robots to inspect the state of a certain surface (e.g., an indoor wall, the surface of a dam, the facade of a building, etc.) to find defects (e.g., holes, fissures, mold, spots, humidity, etc.) as symptoms of potential problems due to, for example, structural imperfections.

In this type of scenario, the aerial robot may operate as an assistant for the human operator who delegates in the vehicle inspection tasks. The robot may have certain inspection abilities (e.g., path planning, defect recognition, etc.). These abilities may reduce significantly the workload of the operator and increase safety, compared to simple manual teleoperation. However, in this type of mission, it is difficult to have robots that operate fully autonomously because they may not have a complete understanding of the environment. Robots may have recognition abilities for certain defects but, sometimes, certain defects are difficult to classify automatically. In this case, the robot may ask for assistance to the operator, which requires a richer interaction model between operator and robot.

The goal of this paper is to present preliminary results of our ongoing research work to analyze more complex human-robot interaction in surface inspection missions. In our work, we have followed the general concept of collaborative control to formulate a specific human-robot interaction model designed for mission inspections. Our approach combines two interaction modes, supervisory and assistance (with a set of interaction patterns). We implemented this model using the software framework Aerostack (`www.aerostack.org`) [1, 2] and developed several flight experiments that proved the adequacy of this approach for aerial robotics.

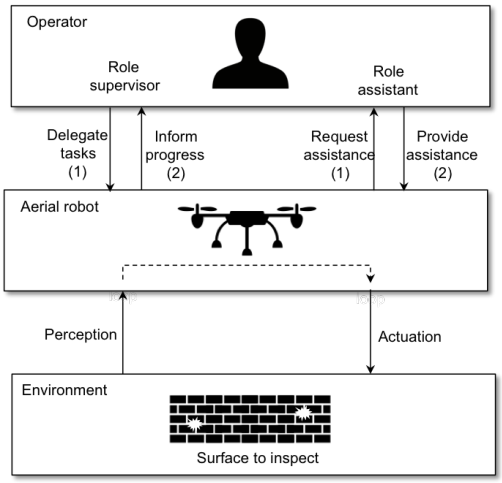


Figure 1: Collaborative control for surface inspection.

The remainder of the paper describes our model and the main results of our work. Section 2 describes the type of user-system collaboration that we have identified for this problem. Section 3 the required inspection abilities. Section 4 describes how we implemented it using Aerostack and, finally, section 5 describes real flight experiments that we performed to refine and evaluate our approach.

2 THE HUMAN-ROBOT INTERACTION MODEL

The problem that we consider in this work is the development of a surface inspection mission performed by a human operator and one or several aerial robots. The goal is to explore a spatial region of a given surface to detect the presence of certain defects. A simple example of this problem is to find imperfections such as fissures or holes in the surface of a wall.

Figure 1 summarizes this type of human-robot interaction. On the one hand, the operator can play the role of supervisor. This form of automation is related to the notion of supervisory control [3] in which a human operator is intermittently acting on the robot to delegate tasks. The robot closes an autonomous control loop through effectors to the environment. This concept has been used to design flexible interaction models, for example, for military mission planning of UAVs [4], swarming networks [5] or remote surveillance system [6].

But, on the other hand, the operator can also play the role of assistant. The human works as a resource for the robot, providing additional information. The robot may ask the human questions as it works, to obtain assistance with perception and cognition. This allows the human to compensate for limitations of autonomy. This is related to the idea of collaborative control in which human and robot work together [7]. The human and the robot dialogue to exchange information, to ask questions, and to resolve differences. This interaction scheme is a kind of mixed initiative approach [8]. Both, the operator and the robot, may take the initiative of the conversation during the dialogue.

Based on this collaborative scheme, we designed a general human-robot interaction model considering messages in categories according to the theory of speech acts [9, 10, 11]. We consider different illocutionary acts to distinguish the intention of the messages, and other subcategories defined by different schemes: DAMSL (Dialogue Act Markup In Several Layers) [12], KQML [13], Move Coding Scheme [14], etc. In particular we use the following categories:

- *Assertive messages*. These messages are sent to give certain information to the receiver (for example, the robot informs the operator the completion of a task).
- *Directive messages*. These messages cause the receiver to take a particular action. Within directive messages, we distinguish between two categories: action directives (requests for action) and information requests.

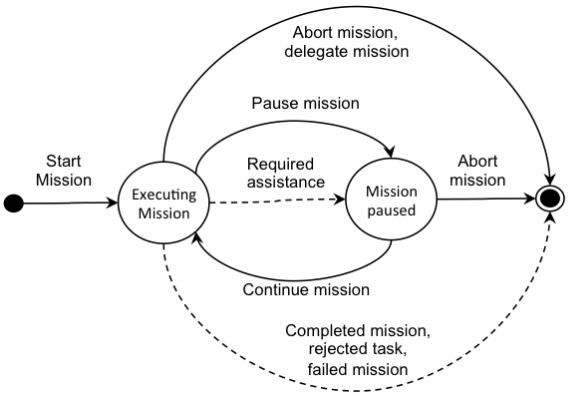
Our interaction model is divided in two interaction modes (supervisory mode and assistance mode) that are described in the following sections.

2.1 Supervisory mode

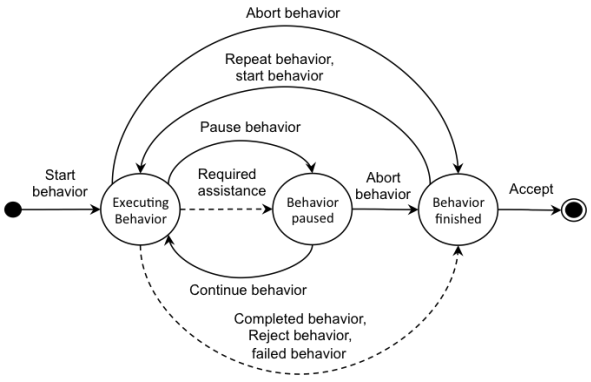
In this interaction mode, the operator sends action directives to the robot in order to delegate mission tasks to the robot. The operator may ask the aerial robot to perform an inspection mission, specifying the area to cover and the exploration strategy. In this case, the relation between operator and robot follows a hierarchical authority (as supervisor-subordinate schema) in which the operator delegates a set of tasks to the robot.

During the development of the mission, the operator observes the robot behavior and the robot sends assertive messages to inform about the mission execution progress (e.g., completed task or finished mission). These messages are useful for the operator to verify that the mission is developing as expected. The operator can interrupt the mission under certain circumstances (for example, to avoid wrong behaviors).

In this interaction mode, the robot shows autonomy to adapt to a dynamic environment while tries to reach its goal [15]. But the robot shows also autonomy to accept or reject the proposed actions according to characteristics of the environment and its own goals (e.g., safety goals) [16]. Its behavior is not completely determined by the influence of the operator.



(a)



(b)

Figure 2: Dialog model for mission control (a) and behavior control (b).

Figure 2a shows the structure of the dialog for mission control. The figure is a state-transition diagram. Transitions with continue arrows correspond to messages sent to the robot by the operator: start mission, pause mission, continue mission, abort mission and delegate mission to other robot. Transitions with dashed arrows correspond to types of messages sent by the robot.

The operator can also control the execution activating and canceling individual robot behaviors (e.g., land, go to point, turn lights on, etc.). Figure 2b shows the dialog model for this type of interaction in which the operator can send messages such as start behavior, pause behavior, continue behavior, repeat behavior and abort behavior.

Both models include a specific transition called required assistance. This transition represents that the robot has reached an impasse because there is an event that prevents from continuing the execution and needs operator assistance.

2.2 Assistance mode

In the presence of uncertainty, a robot may ask the operator for assistance. This interaction mode is required because robots have partial knowledge and are not completely self-sufficient. In this case, the robot works like the field technician (i.e., it is skilled, but may need help) and the operator is like the expert (i.e., she or he can provide assistance when needed) as it is considered in human collaborative control [17].

In the particular case of inspection missions, robots may have recognition abilities for certain defects but, sometimes, certain defects are difficult to classify automatically. In addition, unexpected changes of the environment (e.g., shadows, wind, etc.) may require attention from the operator to decide the appropriate response.

The interaction mode for assistance starts, for example, when the robot is not able to recognize the category of a detected defect in the surface. In this case, the robot sends an information request to ask the operator for the category of the detected defect. The operator answers the category or rejects the detection. In addition, the operator may help the robot proposing motion actions to have better views of the surface.

This interaction mode may also start when the robot is not able to complete a requested task because there is a problem in the environment such as: low visibility, low battery, lost position, high vibrations, impassable barrier, or unstable ground. This includes also the time out event that happens when the robot is not able to complete the task in the expected time due to unknown reasons. The operator helps the robot saying how to respond to these events.

We consider also that the robot may delegate certain specialized tasks to other robots. For example, the robot can transfer part of the mission to other robot because it does not have enough battery charge, or the robot can delegate a certain specialized task that require specialized actuators (e.g., use a special device to mark the detected defect on the wall).

To formulate a model for this type of dialog, we use interaction patterns (see table 1). Each interaction pattern expresses how the robot must interact with the operator in the presence of a certain event. Each pattern is a tuple consisting of two parts. The first part of the pattern is the event that generates an impasse in the development of the mission. The event is the reason why the assistance is required. The second part of the pattern is a list of suggested actions to be presented to the operator (for some of these actions, additional question-answers are needed).

For example, the third pattern in the table is used by the robot in the following way. When there is low visibility, the robot informs about this event to the operator and asks the operator for the next action to do, showing a list of suggested actions. In this case, the operator can select either (1) turn on the lights and continue mission or (2) abort mission and land. If one of these options is selected by the operator, it is performed automatically. The list of suggested actions is not closed. This means that, instead of the suggested actions, the operator can decide to do any other action using the supervisory mode.

Event	Suggested actions
Unknown object	• Zoom in • Zoom out • Learn new category • Horizontal exploration • Vertical exploration
Recognized category	• Confirm category and continue mission • Reject category and continue mission • Learn new category • Mark defect
Problem low visibility	• Turn on the lights and continue mission • Abort mission and land
Problem low battery	• Turn off the lights • Turn off the camera • Delegate mission to another robot and land
Problem time out	• Repeat behavior • Continue mission

Table 1: Example interaction patterns for operator assistance.

In general, the list of actions include different types of actions: (1) behavior control directives (e.g., zoom in, land, etc.), (2) mission control directives (e.g., abort mission, continue mission, repeat behavior), and (3) information requests to the operator (e.g., learn new category, a robot to delegate the mission).

The list of actions should not include more than a small number of options (e.g., 5 options) to facilitate an agile communication with the operator. In addition, the list is ordered according to its probability of selection (most probable action first). This list can be initially given by the programmer. But its components could be also learned and its probability updated according the interaction with the operator.

3 ROBOT BEHAVIORS FOR SURFACE INSPECTION

This section describes the specific robot behaviors that we have considered for surface inspection missions.

3.1 Visual recognition of surface imperfections

One of the requirements of an autonomous robot for surface inspection tasks is the ability to detect abnormal marks in a surface and classify the images in the corresponding category. For this purpose, it is possible to use computer vision algorithms.

In this work, we have used the method based on frequency histogram of connected elements (FHCE) [18, 19]. This method is useful to treat the image pixel by pixel and characterize the different types of flaws of the surface.

This method uses the concept of neighborhood, i.e. for a given pixel $p(i,j)$ of an image, its neighborhood is formed by a set of pixels which distances to p are not greater than two integer values r, s and is defined as $\phi^{r,s}_{(i,j)}$. A connected element is the neighborhood selected such as the intensity I of a pixel is a subset of a given grayscale range $[T-\epsilon,\ T+\epsilon]$:

$$C_{(i,j)}(T) = \phi^{r,s}_{(i,j)} :\ I(k,l) \subset [T-\epsilon,\ T+\epsilon],\ \forall (k,l) \in \phi^{r,s}_{(i,j)}$$

Given the previous definitions, $H(T)$ is defined as the sum of all the connected elements for each pixel of an image on different gray levels T, where T is greater than 0 and inferior than the maximum intensity minus one:

$$H(T) = C_{(i,j)}(T),\ 0 \leq T \leq I_{max} - 1 \qquad (1)$$

We use this algorithm to separate the flaws from the background, dividing the process into several steps. We defined the neighborhood shape with a square kernel to fit both the hole and fissure needs. Our square kernel is formed by a 3×3 matrix (i.e. $r = 1$ and $s = 1$) of neighboring pixels where the center pixel is the target.



Figure 3: Example of original vs resulting image after flaw separation.

To find the connected elements in the kernel, we calculate the standard deviation of the grayscale values between pixels within the same neighborhood. Then, given a threshold th (in our case 0.1), if the standard deviation is inferior than th, we assign to the pixel the mean gray value of all the pixels of the kernel. If not, we preserve its original grayscale value.

Then, to calculate FHCE we apply equation 1 to the resulting image after separating connected elements. According to FHCE result, we see that the region related to a flaw is characterized by a range of gray levels between 1 and 50 (i.e. dark gray). If we assign a white value to all the pixels in that range, we have as a result an image with all the flaws separated from the background (Figure 3).

Finally, we have to distinguish between categories of flaws. In this work, two main flaws were considered on the inspected surface: fissures and holes. Fissures can be characterized as a linear flaw, which is why we search for areas where white values are concentrated along a certain linear direction. Holes have to be defined as a square (or circular) flaw, which is why we search for areas where white values are concentrated along a square area of a certain dimension. As a result, we will get delimited areas for both flaw types (see Figure 4).

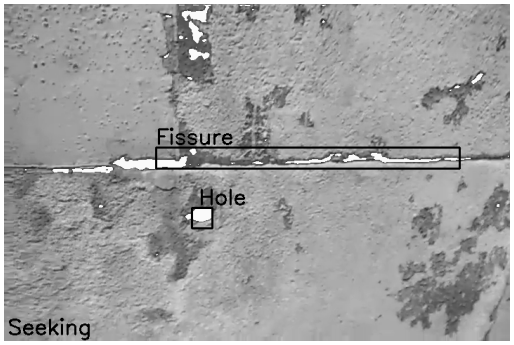


Figure 4: Example of image classification.

3.2 Motion behaviors

In addition to visual recognition of flaws, the robot requires specifc motion behaviors to develop a safe exploration search with an appropriate degree of autonomy. Table 2 shows a list of behaviors that the robot may need for inspection tasks.

We have designed this set of behaviors to be reusable for different inspection missions. There are general behaviors (e.g., TAKE OFF, LAND) and more specific behaviors for inspection tasks (e.g., ZOOM, EXPLORE). For example, the behavior EXPLORE develops an exploration search following a prefixed trajectory to cover the complete surface. The trajectory may follow different strategies such as vertical search or horizontal search (see Figure 5).

4 SYSTEM IMPLEMENTATION

We implemented an experimental software prototype to refine and evaluate the approach for surface inspection missions described in this paper. For this purpose, we used and extended the software framework Aerostack (`www.aerostack.org`) [1, 2]. Aerostack is a general software framework for aerial robotics that provides different soft-

Behavior	Description
TAKE OFF	The robot takes off
EXPLORE	The robot explores the surface following a search strategy. Argument: strategy VERTICAL, HORIZONTAL
LAND	The robot lands
GO HOME	The robot returns to the home base
ZOOM	The robot moves forward to zoom in or backwards to zoom out. Argument: IN, OUT
MOVE	The robot moves a prefixed distance (e.g., 0.5 m) in the specified direction. Argument: direction UP, DOWN, RIGHT, LEFT
KEEP HOVERING	The robot pauses its movement
TURN LIGHT	The robot turns the light on or off. Argument: ON, OFF
MARK SURFACE	The robot draws a colored circle around the flaw
APPROACH	The robot goes to the position of another robot. Argument: robot identifier.

Table 2: Example behaviors for inspection tasks.



(a) (b)

Figure 5: Example of exploration strategies: (a) horizontal and (b) vertical.

ware components and a combination scheme for building the software architecture for autonomous operation of an aerial robotic system. For example, Aerostack provides software components with perception algorithms, SLAM algorithms, controllers, mission plan interpretation methods, multi-robot communication and a general graphical user interface for human-robot interaction.

Figure 6 shows a block diagram with the main software components of our implementation. In the figure, blue blocks correspond to processes provided by Aerostack and orange blocks are new processes that we programmed and added to Aerostack for inspection problems. For instance, we programmed the process called surface defect recognizer that implements the FHCE algorithm presented previously, using the images captured by the front camera of the drone. In this implementation of FHCE we also used simple routines provided by the OpenCV library (e.g., for representation of images, pixel manipulation, etc.). Aerostack provides a library of behaviors such as general motion behaviors (take off, land, got to point, etc.). We extended this library with specific

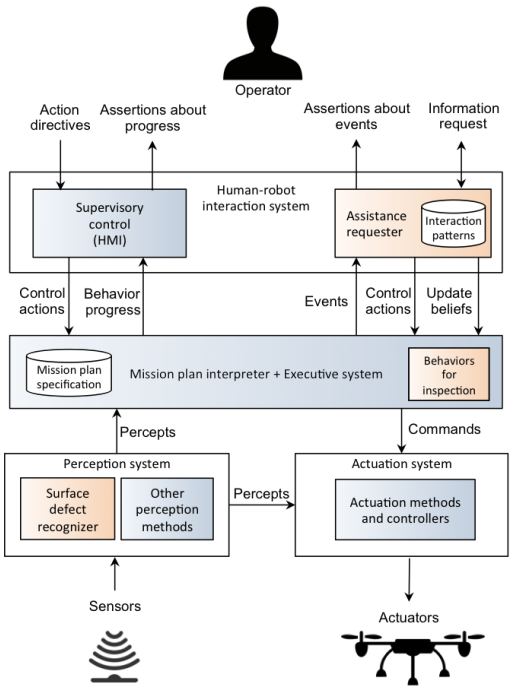


Figure 6: Block diagram with the main software components.

behaviors useful for inspection missions. For example, we implemented the behavior explore with different exploration strategies, or the behavior zoom, to move closer or farther to the surface.

The dialog between the operator and the robot uses three communication channels. First, Aerostack provides a graphical user interface (GUI) to interact with the drone. This interface allows the operator to do supervisory control. Figure 7 shows an example of window presented by the Aerostack GUI which can be used by the operator to control the mission execution (start mission, abort mission, etc.). This window is called control panel, with specific buttons and fields, where operator can control the state of the mission. The robot uses also this panel to show the current action (e.g., take off, go to point, land, etc.). This is useful to help the operator to supervise the correct execution of the mission.

The Aerostack GUI also provides another channel that the operator can use to send behavior directives to the robot. In this case, the operator selects the specific behavior to do (e.g., land, take off, etc.).

The version of Aerostack that we used for this work did not provide the assistance request for human-robot interaction, as we defined in this paper. Therefore, we programmed a new process, called assistance requester, to provide this service. This process receives events corresponding to a mission impasse (e.g., the presence of an unrecognized image or the presence of a problem) and interacts with the operator requesting the appropriate information according to the inter-

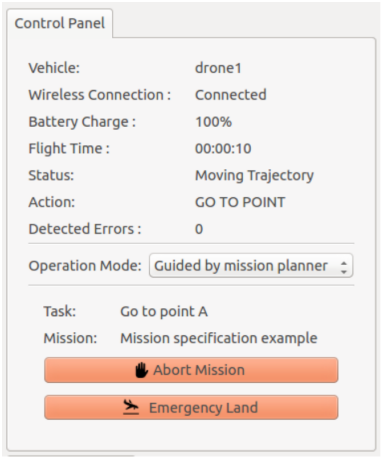


Figure 7: Window for mission control provided by Aerostack.

action patterns.

We programmed the assistance requester using a command line interpreter with simple commands with two letters (e.g., cm for continue mission, etc.). However, for a future more complete implementation, we designed two alternative options for assistance request. The first option is based on a interruption window presented to the operator (a pop-up window) asking for operator attention and requesting an answer. Figure 8 shows an example of the design of such a window. We asume that while this window is presented, the robot keeps hovering and, if the operator does not answer after a limited time T (e.g., $T = 45$ seconds), the robot continues the mission.

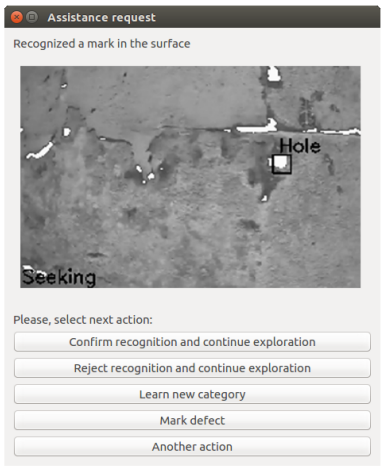


Figure 8: Example window presented to the operator for assistance request.

An alternative option for the previous window is based on using voice messages and speech recognition used in natural user interfaces (as we propose in [20]). In this case, the robot generates the following text message:

> *The mission is stopped because I have recognized the mark shown on the monitor. Please, say what to do next: (1) confirm recognition and continue exploration, (2) reject recognition and continue exploration, (3) learn a new category, (4) mark defect, and (5) do another action.*

5 EXPERIMENTS

This section shows a preliminary set of scenarios that we analyzed to evaluate our interaction model. Table 3 shows the set of flight experiments that we considered to cover different situations of interaction. The experiments illustrate the kind of collaborative human-robot interaction presented in this paper.

Scenario	Description
Low visibility and mission delegation	Robot R1 finds a dark area which does not allow the proper recognition. The operator orders R1 to turn on the light and continue with the mission. Then, the drone has a low battery charge. The operator transfers the mission to robot R2.
Distributed specialized tasks	Robot R1 finds a hole. The operator orders to delimitate the hole zone. The drone asks for painter drone help. Painter drone draws a circle around the hole. The first drone finishes the mission.
Reconstruction of large fissure	Robot R finds a large fissure. The operator orders to change the inspection strategy to up/down. The drone starts moving taking several pictures of the fissure. Once finished, it makes a reconstruction of the fissure and classifies it correctly.
Zoom out for large fissure	Robot R finds a large fissure. The operator orders to zoom out. The robot gets a better (complete) view and classifies it correctly.
Lost position	Robot R loses its position with respect to the wall. The operator relocates the drone in the inspection area using movement orders and orders to continue the mission.
New object to recognize	Robot R finds an unknown imperfection that cannot be classified following the trained imperfections. The drone asks the operator what to do, and the operator decides to create a new class stain.

Table 3: Example scenarios corresponding to flight experiments.

For example, in the third scenario, the robot develops the inspection until it finds a large fissure. The robot recognizes an unknown object but it is not able to classify it. Then, the operator orders to zoom out. Then, the drone zooms out from the fissure to get a better (complete) view and then, classifies

correctly the defect. Then the drone continues the mission and no more objects are recognized.

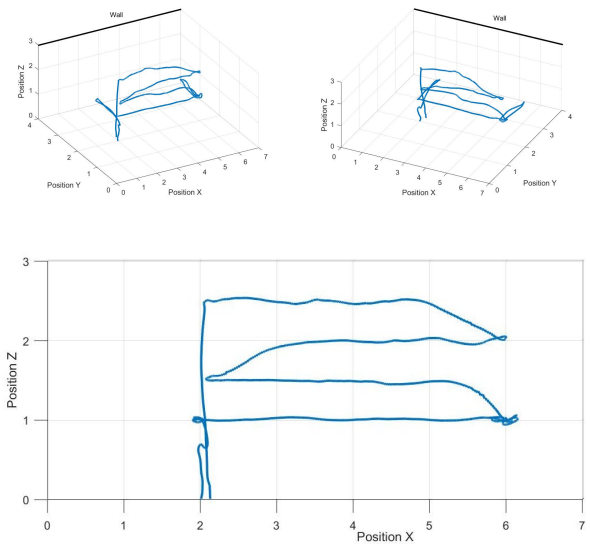


Figure 9: Example exploration trajectory followed by the robot.

Figure 9 shows the trajectory developed by the robot in one of the flight experiments. This example corresponds to an aerial platform AR Drone 2.0 (Figure 10) performing a indoor inspection to find holes and fissures on a wall. The whole trajectory is covered in 1 minute and 19 seconds. The robot develops autonomously an horizontal exploration starting from the takeoff point $(2,\ 2,\ 0)$ and landing at the same point after the exploration. There are two points where the robot zooms in and out to have closer views of the wall: point $(2,\ 3,\ 1)$ and point $(6,\ 3,\ 1)$. In this particular example, the robot performs an autonomous inspection without any interaction with the operator.



Figure 10: The aerial robot (AR Drone 2.0) during wall inspection.

Figure 11 shows examples of recognized defects on the wall, corresponding to a fissure and a hole. In general, the experiments proved that the recognition method was able to classify correctly the 88.5% of fissures and the 49.5% of the holes (evaluated with a sample of 200 images). The recognition performance can be acceptable for the interaction goal presented in this paper, where the robot requires the confirmation of the operator. Especially, the algorithm gives good results on well differentiated flaws from the background. A high performance of the recognition process was outside the goal of this work and may be achieved with future research work.

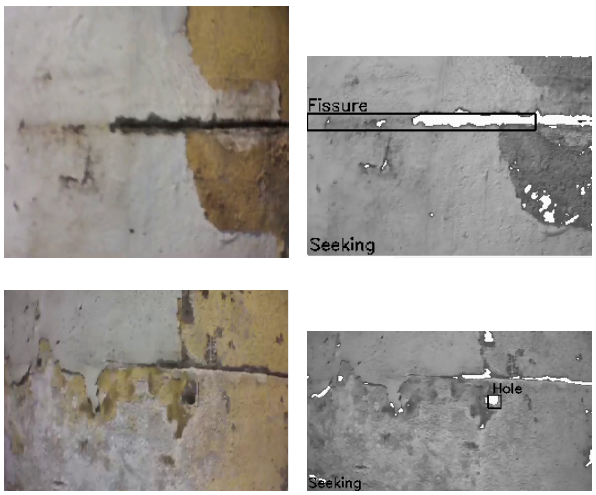


Figure 11: Example of recognized defects on the wall.

6 CONCLUSIONS

In this paper we have presented our human-robot approach for surface inspection tasks based on a collaborative control with a flexible dialogue between operators and robots with a mixed initiative interaction. The human-robot dialog scheme is defined with two main interaction modes, supervision and assistance, with generic interaction patterns. The presented model is generic to be applied for different surface inspection missions with one operator and one or more aerial robots.

Compared to a conventional tele-operated inspection system, the solution presented in this paper leaves more autonomy to the aerial robot, which can reduce the cognitive load of operator during the mission development.

We developed an experimental prototype of a software system to refine and validate this model using the framework Aerostack. The experiments proved that this approach can be used in aerial robotics for surface inspection missions (e.g., vision based indoor missions). However, this preliminary design and implementation still needs to be extended with additional components and more extensive evaluation to show in more detail its contributions and its practial utility.

We plan to extend this approach in the future with more types of behaviors (e.g, other complex exploration strategies) and other types of categories of defects to be used in related scenarios.

ACKNOWLEDGEMENTS

This research work has been partially supported by the Spanish Ministry of Economy and Competitiveness through the project VA4UAV (Visual autonomy for UAV in Dynamic Environments), reference DPI2014-60139-R.

REFERENCES

[1] J. L. Sanchez-Lopez, R. A. Suárez Fernández, H. Bavle, C. Sampedro, M. Molina, J. Pestana, and P. Campoy. Aerostack: An architecture and open-source software framework for aerial robotics. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 332–341, June 2016.

[2] J. L. Sanchez-Lopez, M. Molina, H. Bavle, C. Sampedro, R. A. Suárez Fernández, and P. Campoy. A multi-layered component-based approach for the development of aerial robotic systems: The aerostack framework. *Journal of Intelligent & Robotic Systems*, May 2017.

[3] T. B. Sheridan. *Telerobotics, automation, and human supervisory control*. MIT press, 1992.

[4] C. A Miller and R. Parasuraman. Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control. *Human factors*, 49(1):57–75, 2007.

[5] M. L. Cummings. Human supervisory control of swarming networks. In *2nd annual swarming: autonomous intelligent networked systems conference*, pages 1–9, 2004.

[6] J.-S. Lee, M.-C. Zhou, and P.-L. Hsu. An application of petri nets to supervisory control for human-computer interactive systems. *IEEE Transactions on Industrial Electronics*, 52(5):1220–1226, 2005.

[7] T. Fong, C. Thorpe, and C. Baur. Multi-robot remote driving with collaborative control. *IEEE Transactions on Industrial Electronics*, 50(4):699–704, 2003.

[8] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166. ACM, 1999.

[9] J. R. Searle. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press, 1969.

[10] J. R. Searle and K. Gunderson. Language, mind, and knowledge. *Minneapolis: University of Minnesota*, 1975.

[11] J. L. Austin. *How to do things with words*. Oxford university press, 1975.

[12] M. G. Core and J. Allen. Coding dialogs with the damsl annotation scheme. In *AAAI fall symposium on communicative action in humans and machines*, volume 56. Boston, MA, 1997.

[13] Y. Labrou and T. Finin. A proposal for a new kqml specification. Technical report, Technical Report Technical Report TR-CS-97-03, University of Maryland Baltimore County, 1997.

[14] J. Carletta, S. Isard, G. Doherty-Sneddon, A. Isard, J. C Kowtko, and A. H Anderson. The reliability of a dialogue structure coding scheme. *Computational linguistics*, 23(1):13–31, 1997.

[15] R. Murphy. *Introduction to AI robotics*. MIT press, 2000.

[16] C. Castelfranchi. Guarantees for autonomy in cognitive agent architecture. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 56–70. Springer, 1994.

[17] T. Fong, C. Thorpe, and C. Baur. Robot as partner: Vehicle teleoperation with collaborative control. In *Multi-robot systems: From swarms to intelligent automata*, pages 195–202. Springer, 2002.

[18] D. Maravall and M. A. Patricio. Image segmentation and pattern recognition: a novel concept, the histogram of connected elements. In *Pattern recognition and string matching*, pages 411–463. Springer, 2003.

[19] M. A. Patricio and D. Maravall. A novel generalization of the gray-scale histogram and its application to the automated visual measurement and inspection of wooden pallets. *Image and vision computing*, 25(6):805–816, 2007.

[20] R. A. Suárez Fernández, J. L. Sanchez-Lopez, C. Sampedro, H. Bavle, M. Molina, and P. Campoy. Natural user interfaces for human-drone multi-modal interaction. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1013–1022, June 2016.

An Intelligent Unmanned Aircraft System for Wilderness Search and Rescue

Huai Yu*, Shijie Lin, Jinwang Wang, Kaimin Fu, Wen Yang
School of Electronic Information, Wuhan University, No.299 Bayi Road, Wuhan 430072, China
CETC key laboratory of aerospace information applications, Shijiazhuang 050081, China

ABSTRACT

In this paper, we presented a wilderness search and rescue (WiSAR) system based on DJI M100 Unmanned Aerial Vehicle (UAV) and a ground station to search and rescue the survivors in wild. We combined infrared and optical target detection to increase the detection speed and accuracy and used multiple sensors to make this system can autonomous avoiding obstructions and landing on mobile platform. For further increase the Average Precision of SSD, we build a field people dataset UAV-PP and use ResNet-101 as the base net. The actual flying test have been conducted in multiple situations to verify the feasibility of our WiSAR system. Our WiSAR system laying a solid foundation for building a more intelligent search and rescue system based on UAV.

1 INTRODUCTION

Wilderness search and rescue (WiSAR) is very necessary and difficult due to its vast territory and frequent field disasters. Generally, WiSAR is racing with time, every second counts. Search and rescue operations usually need a lot of manpower and resources. Traditional rescue methods, like human-base search, are inefficient and can easily miss the best rescue time. In recent years, the rapid development of Unmanned Ariel Vehicles (UAV) provides another better way for rapid search and rescue. UAV equipped with image acquisition cameras and a variety of sensors, transport the obtained video to the ground station. In addition, the UAV is agile, flexible, and can perform actions that are difficult to perform by humans. These features make UAV more suitable for WiSAR. However, currently UAV in WiSAR mainly uses the image acquisition module, cares little about the automation, the application scenarios are relatively limited. Therefore, the key technologies in the design of UAV for WiSAR are studied, i.e., the autonomous obstacle avoidance, path planning and automatic landing. Meanwhile, we utilize the target detection and recognition technologies to efficiently detect and locate survivors.

*Email address: yuhuai@whu.edu.cn



Figure 1: The hardware about WiSAR system.

2 WISAR SYSTEM DESCRIPTION

The hardware of our WiSAR system is shown in Figure 1.The whole system can be divided into three components, i.e., drone system, ground station and remote control. The details about the three parts are listed as follows,

- Drone system consists of a DJI Matrice 100 drone with an on-bard PC Maniford, five stereo-vision sensors to provides around and downside depth cloud maps, two ultrasonic sensors to keep a safety height, a DJI Zenmuse X3 camera to provide optical images, a FLIR VUE Pro to provide infrared images.
- Ground work station is a DELL laptop with a very powerful GPU NVIDIA Quadro M5000, which is mainly used to process images.
- The remote control is a DJI standard remote control with an Android phone.

Our WiSAR system works as: after the whole system is powered on, the on-bard computer Manifold starts to take control and guide the drone autonomously take off. Then the drone flies follow the global setting path. When there are obstacles in the front or downside (point clouds from the stereo-vision sensors), the Robot Operating System (ROS) [1] (Robot Operating System) navigation package is used to set the local path. Meanwhile, a searching command is sent from the APP in remote control module to the drone by 2.4GHz/5.8GHz

wireless communication module. The on-bard computer begins screening the infrared video and sending ROI (Region of Interest) to remote control. When the ground station receives the searching command and ROI, the improved-SSD algorithm is used to detect target using the optical video stream transported from the remote control. The data link is shown in Figure 2.

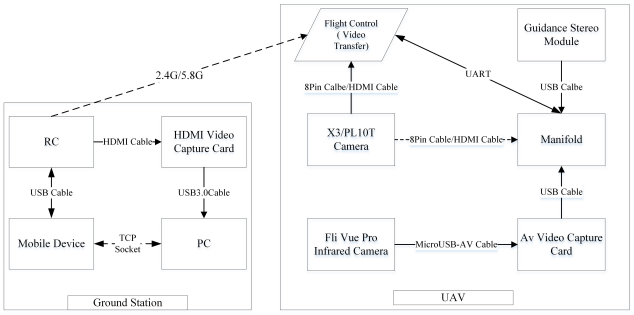


Figure 2: The data link diagram of Our WiSAR system.

3 People Search and Rescue

Finding lost people is the main job of our WiSAR system. In this section, we provide a method to reduce the false alarm of detection by combining infrared detection with optical detection. We use the infrared images for assistance. We use morphological methods to obtain the salient region, then converted it into the target candidate regions. After that, we align the infrared image and the optical image. We use the improved SSD [2] model to real-time detect people, based on the candidate region.

3.1 Infrared image target detection

Usually, living target, like humans and other warm blooded animals, can be distinctly displayed on infrared images because these targets can radiate more energy than the background. In our WiSRA system, we using FLIR VUE Pro, a thermal imager with 640×512 pixels resolution, to effectively detect the thermal information within hundred meters away. When our WiSAR system fly in the air, the resolution of living targets is usually 30×30 pixels. This relative low makes the information easily lost and other disturbances make the analysis of infrared image difficult (Figure 3a). So we apply several methods described below to handle the infrared images first, making it easy to process.

First, it is necessary to carry out a equalization of the image because strong light will cause the temperature of the water in the air rising, radiating more infrared rays that can affect thermal imager. While the distribution of water vapor is usually evenly distributed in a smaller area, the infrared energy radiated outward is also evenly distributed. Let the length of the image I be W, the width is H, $P(x, y)$ is the gray level of the position (x, y), $P'(x, y)$ is the gray value of (x, y) after the mean translation, then we have:

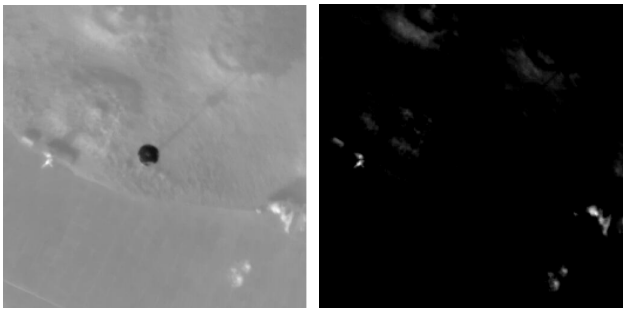
$$P'(x,y) = P(x,y) - \frac{1}{WH}\sum_{i,j\epsilon W,H} P(i,j) \qquad (1)$$

After equalization, we use gray-scale transformation to let the gray scale range from 0 to 255 value instead of negative value. The linear gray-scale transformation function is define as,

$$D_b = f(D_a) = kD_a + b \qquad (2)$$

where k is the slope, b is the intercept. D_a indicates the grayscale of the input infrared image. D_b indicates the grayscale of the output image.

When the gradient k is greater than 1, the contrast of the processed image will increase, and if the gradient k is less than 1, the contrast will decrease. Here, we highlight the target by increasing the contrast. By limiting the grayscale, the gray values of the infrared images can be distributed equally in different lighting conditions, allowing for further processing. By doing this, the significant region (temperature anomaly region) is effectively extracted (Figure 3b).



(a) (b)

Figure 3: Infrared image gray scale processing. (a) Blurry infrared image contained mutiple targets. (b) Image after equalization and gray-scale transformation.

Further, we use the Top Hat transformation to reduce the impact of light. Considering that the target size under the UAV perspective is small and the brightness in the image is high, Using the top hat transformation is reasonable.

From the previous steps, we have been able to get less noise results. However, the background of the wild environment is very complex, just using infrared search is unable to achieve the reliable results. Therefore, if the number of candidate regions, obtained by the above image analysis process, is greater than 0 (Figure 4), the UAV will transmit report instructions and the coordinates of candidate regions to the ground station. The ground station will automatically use optical detection to search people based on these informations.

3.2 Optical image taget detection

Recently, the state-of-the-art deep conventional neural network (DCNN) frameworks like YOLO9000 [3] and SSD

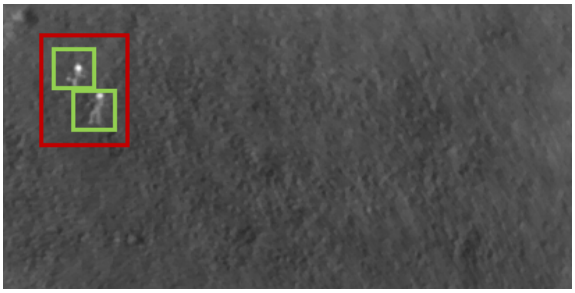


Figure 4: Candidate region detected by thermal camera.

[2] have been presented. Considering the past target detection methods based on statistical learning, such as HOG [4] and DPM [5], are used artificial designed features, not enough for field complex scenes. And field scenarios, people's contours, edges, textures may not be very clear, and even there will be a certain degree of distortion (Figure 5a and 5b). Thus, we decide to use the state-of-the-art methods to overcome the problems above, that is DCNN. Through the use of our own field people dataset UAV-PP, our WiSAR system can effectively locate the field people. Based on the requirements for real-time and high accuracy, we compared multiple DCNN frameworks and finally decided to adopt the improved-SSD framework.



(a) (b)

Figure 5: People under UAV perspective. (a) People in different postures, (b) People with partial occlusions and without partial occlusions.

The network structure of SSD [2] is very clear and easy to understand. First it uses a forward-propagating CNN as a base network. This forward convolution network can produce a series of fixed-size brackets and these enclosing boxes contain scoring of various categories of objects. And then it uses Non-maximum Suppression (NMS) to filter out the final prediction results. The feature maps, from the same network while at different levels, have different size of receptive field. However, the SSD model does not have to let the default bounding box corresponding to the receptive field in every layer, but let the feature map to be responsible for the prediction of a particular scale. Suppose you want to use m feature graphs to predict. Then the scale of the default bounding box corresponding to each feature map s_k can be calculated according to the following formula:

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m-1}(k-1), k\epsilon[1,m] \quad (3)$$

We changed the scale partition, making m equal to 4, s_{min} equal to 0.4 and s_{max} equal to 0.8, because the scale of the target in UAV perspective does not change much. In addition, we increased the batch size from 128 to 512, because the actual size of the target is relatively small.

The original base network of SSD uses the prediction layer in front of VGG16 [6]. VGG16 model has been proved to be a good classification prediction model, but its structure is too complex and with many layers, a 32×32 target after VGG becomes 2×2 size, makes the extra layer is easy to lose semantic information. So the original SSD for small size targets (such as UAV view of the people) detection effect has been affected. Therefore, for the small target detection, it is necessary to increase the semantic information of the context[7]. Residual Network(ResNet)[8] proposed by He et al can preserve as much target semantic information as possible. ResNet introduces a shortcut between the output and the input (shortcut), that is identity function, rather than a simple stack network. This can solve the problem that the semantic loss occurs due to the network being too deep. Allowing people to further increase the depth of the network. So we use the depth of the ResNet-101 structure as the SSD base network to improve. It is worth noting that even using ResNet-101 will not significantly increase the time required for target detection. Through the above improvements, SSD-ResNet model for small target detection has been improved.

3.3 Combination of infrared and optical detection

When trying to combine the optical and infrared target detection, we need some coordinate transformation, because the optical camera and the infrared camera have a certain physical distance during installation . In order to get the correct candidate region coordinate mapping, we need to get the conversion relationship between the two camera coordinate systems. In Figure 6,

θ_1 is the viewing angle of the infrared camera,

θ_2 is the viewing angle of the optical camera (taking the x-axis as an example),

The point $p_f(u_f, u_f)$ in the infrared camera image coordinate system will be mapped to the image coordinates $p_p(u_p, u_p)$ of the optical camera by the following transformations.

$$u_p = u_{po} + WIDTH_P * (u_f - u_{fo'})/WIDTH_F + \frac{d}{f_x} \quad (4)$$

$$v_p = v_{po} + HEIGTH_P * (v_f - v_{fo'})/HEIGTH_F + \frac{d}{f_x} \quad (5)$$

Where,

$WIDTH_P$ is the width of the optical image,

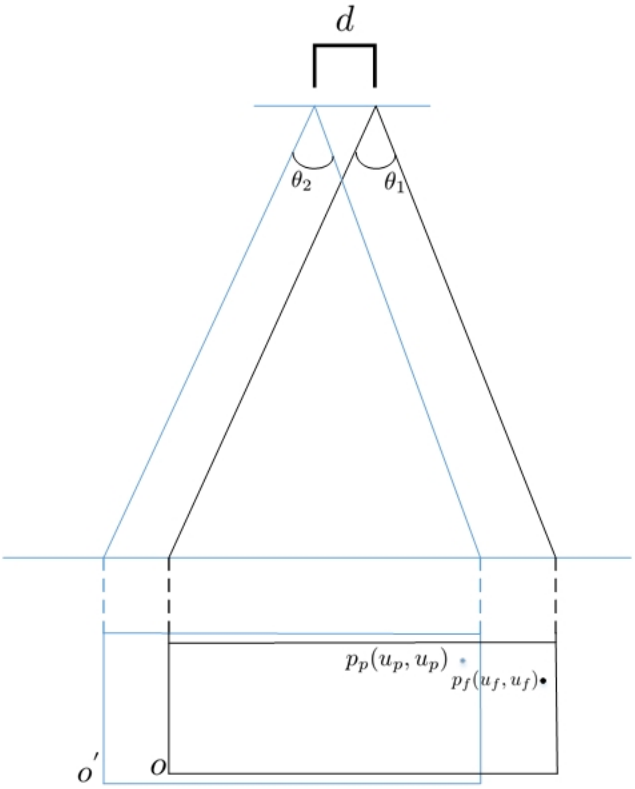


Figure 6: Relationship between infrared and optical camera coordinate system.

$WIDTH_F$ is the width of the optical image,
$HEIGTH_P$ is the height of the infrared image,
$HEIGTH_F$ is the height of the infrared image,
d is the installation position deviation, in our WiSAR is 6 cm,
f_x is actual physical length of each pixel in an optical camera.

By using two of the above image coordinate transformation equations, it is possible to obtain the corresponding position of the target candidate region on the optical image. Then set it as the region of interest (ROI) in optical images and perform optical target detection based on this ROI.

4 Autonomous Obstacle Avoidance and Landing

As for autonomous obstacle avoidance, we adopt the obstacle avoidance scheme based on binocular vision. It mainly uses the parallax principle to carry out motion estimation, so as to obtain the depth information of the obstacle in front of the UAV. And then update the cost map based on the point clouds, and further update the obstacle avoidance route according to the cost map, so as to achieve the purpose of avoiding obstacles [9]. If the UAV encounters an uneven terrain like Figure 8, the ultrasonic detector mounted on the bottom



Figure 7: Human detection results using our WiSAR system.

of the UAV will initiative send signal to keep a safety flight altitude. The results of the obstacle avoidance experiments for static obstacles (wall, tree, etc.) and moving obstacles (pedestrian) show that the vision avoidance scheme based on binocular vision is more reliable than Light Detection And Ranging (LIDAR).

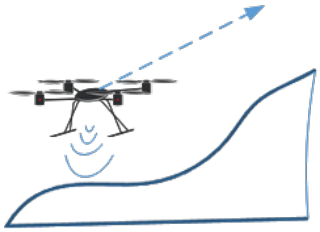


Figure 8: UAV use ultrasonic to avoid obstruction .

Regarding the autonomous landing, we introduce the visual fiducial system AprilTag [10] from the Augmentation Reality filed. By encoding an AprilTag, the detection algorithm can greatly reduce the false alarm, and because of the introduction of fault tolerance mechanism, the miss rate is also maintained at a very low level. After detecting the mark, the UAV can dynamically adjust its pose to achieve accurate landing on the mobile platform by solving the PnP problem. Experimental results show that the position deviation of landing is less than 15 cm, which meets the requirement of real landing scenario where a UAV is likely to land on a platform with limited area, such as roof and truck rear.

AprilTag information is only expressed in black and white blocks, without any symmetry(Figure 9a). In our WiSAR system, we use the 25h9 set. First, the useful contour information is extracted by image graying(Figure 9b) and used the adaptive binarization algorithm to binarize gray scale images (Figure 9c), and then processed by Gaussian filtering (because Gaussian filtering can better preserve the edge information [11]), and finally through the basic and further screening (preventing false contours detection in internal AprilTag), we can get the correct results (Figure 9d). Further, we introduced the

Hamming code [12] to improve error correction and screen out false alarms, making our AprilTag looks like Figure 9e.



(a) (b) (c) (d) (e)

Figure 9: Detection process of AprilTag. (a) Original image with 6 AprilTags, (b) Image after gray processing, (c) Image after binarization, (d) AprilTag edge detection, (e)AprilTag after Hamming coding.

5 Experiments and Results

5.1 Field people dataset UAV-PP

We build and put forward a field people dataset UAV-PP and used the VOC [13] dataset format to increase the versatility of UAP-PP dataset. Because the people under the UAV perspective are very different with the people under the ground camera perspective, and the deep learning is highly rely on large volume of data. So lacking UAV perspective people database will heavily influence the average precision of detection.

UAV-PP including 3180 original images, resolution of these images are 1000×1000 pixels, each image contains 10 to 20 targets (people), each positive sample has a corresponding ground truth bounding box. We provide a labeling tool (Voc-Annotation-Tool) to facilitate the placement of Ground Truth. It has the ability to batch rename pictures, import pictures, and mark the image in VOC format and generate the corresponding structure files. The software runs as Figure 10.



Figure 10: Voc-Annotation-Tool.

To improve the robustness of the DCNN model, we mainly use side view and bottom view samples. The ratio of the human training images in the various postures are about 1: 1: 1 (bottom view $85°$, side view $45°$, and other gestures respectively). Figure 11 shows the example targets in UAV-PP dateset.



Figure 11: Human sample images.

5.2 Comparison of target detection algorithm

In the same test platform (as described in Table 1), we used Average Precision (AP) to compare different target detection frameworks and found that:

Table 1: Algorithm test platform

CPU	Intel E3-1505M
GPU	NVIDIA Quadro M5000, 8GB
RAM	64GB

DPM has a high demand for image capture quality and perspective. In the different perspectives of people, the DPM algorithm can not extract the characteristics from numerous samples. DPM received 63.82% AP on the test set in the UAV-PP dataset, with an average speed: 3.0 Seconds per image.

As a successful deep learning framework, the detection effect of Fast-RCNN is quite good. But due to the various viewing angles, there are still some missed targets can't be detected. Using Imagenet dataset, learning rate α equal to 0.0003, batch size equal to 128, after 60000 times iterations, we have 72.47% AP of Fast-RCNN, the average speed: 0.16 Seconds per image.

SSD is faster and more accurate than Fast-RCNN when using the same dataset (ImageNet [14] dataset) and test set. Average speed of SSD: 0.16 Seconds per image. AP of SSD: 80.85%.

After that, we changed the original SSD base network to the residual network, and optimized the parameters of SSD, and got 88.92% AP. Meanwhile, the average speed did not increase, which still was 0.08 seconds per image. The comparisons of AP and speed are shown in Figure 12 and Figure 13.

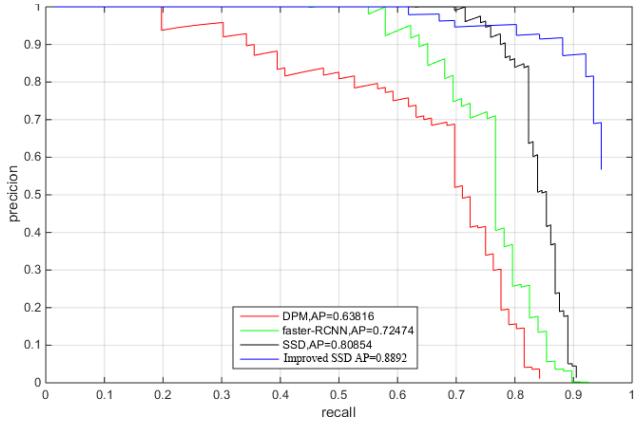


Figure 12: Average Precision between different target detection algorithms.

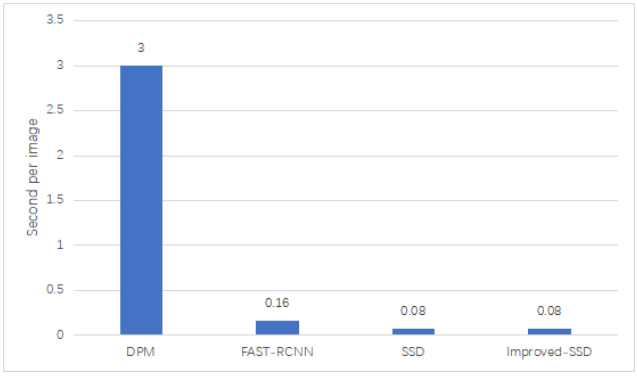


Figure 13: Speed between different target detection algorithms.

5.3 *WiSAR system test*

The whole WiSAR system is a multi-module integrated system. For the UAV side, data processing and instruction transmission is mainly in ROS. For the ground station side, the system includes a supporting demonstration system to satisfy the practical need. Which includes an Android application (Figure14) and a PC application (Figure15).

First of all, we conducted an independent obstacle test, the results show that the UAV can effectively avoid moving pedestrians and other obstacles. Then we carried out 20 tests to test the autonomous landing (Figure 17a and 17b). Let the center of the platform as the origin point,we have the results that: the average deviation of X axis is 9.2 cm, the average deviation of Y axis is 9.0 cm. The results are enough to meet the requirements of landing on a moving truck. Finally, we use 15 tests to test the time and distance flying ability of our WiSAR system when under full load. The average farthest flight distance is 4.2 km at full load and the average maximum flight time is 15 minutes and 10 seconds.

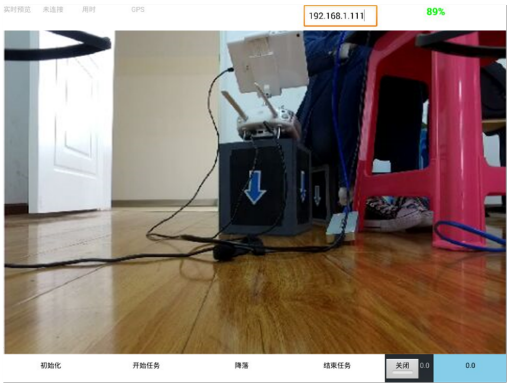


Figure 14: Android application interface of our WiSAR system.

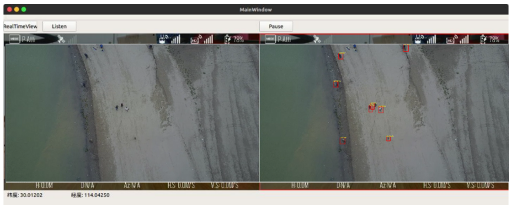


Figure 15: PC application interface of our WiSAR system.

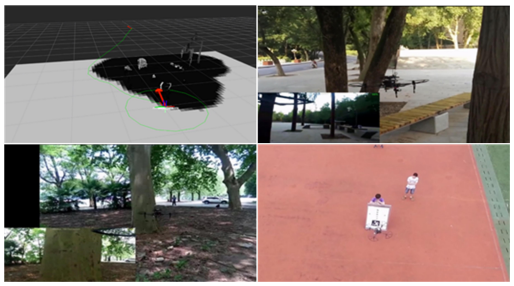


Figure 16: Some results of obstacle avoidance.



(a) (b)

Figure 17: Autonomous landing test. (a) Autonomous landing on truck rear, (b) Autonomous landing on simulated truck rear.

6 Conculsion

In conclusion, we have completed the design, prototyping and construction of our WiSAR system. The system adopts modular development strategy, leading to low coupling degree and high portability. The accompanying software applications include an Android application and a multi-threaded graphical PC application. Both applications respond quickly and interactively. The experimental results show that the performance of our WiSAR system is fully functional. The simulated search and rescue tasks can be successfully accomplished, which lay a solid foundation for building a more intelligent search and rescue system based on UAV.

Acknowledgment

The research was partially supported by the CETC key laboratory of aerospace information applications under Grant KX162600018.

References

[1] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, 2009.

[2] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*. Springer, 2016.

[3] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*. IEEE, 2017.

[4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*. IEEE, 2005.

[5] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*. IEEE, 2008.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICRL*, 2015.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*. Springer, 2014.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*. IEEE, 2016.

[9] Lionel Heng, Dominik Honegger, Gim Hee Lee, and et al. Autonomous visual mapping and exploration with a micro aerial vehicle. *Journal of Field Robotics*, 31(4):654–675, 2014.

[10] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *ICRA*. IEEE, 2011.

[11] Guang Deng and Laurence Cahill. An adaptive gaussian filter for noise reduction and edge detection. In *NSS/MIC*. IEEE, 1993.

[12] Chi-Shiang Chan and Chin-Chen Chang. An efficient image authentication method based on hamming code. *Pattern Recognition*, 40(2):681–690, 2007.

[13] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, jun 2010.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*. IEEE, 2009.

A honeybee's navigational toolkit on Board a Bio-inspired Micro Flying Robot

Erik Vanhoutte, Franck Ruffier and Julien Serres *
Aix-Marseille Univ, CNRS, ISM, Marseille, France

Abstract

In this paper, a 395-gram micro flying robot equipped with an insect-inspired visual system is presented. The robot's visual system was designed to make it avoid both ground and lateral obstacles, using optic flow-based regulation principles. The quadrotor is an open-hardware X4-MaG drone with an active gimbal system based on a pair of serial servo motors, which stabilizes 8 retinas dedicated to optic flow measurements in the 25 $^\circ/s$ to 1000 $^\circ/s$ range, each of which comprises 12 auto-adaptive pixels working in a 7-decade lighting range. The X4-MaG drone is tested in front of a slanted wall, its quasi-panoramic bio-inspired eye on board is able to estimate the angle of incidence in the 0° to 50° range with an error of less than 2.5° when flying. These experimental results are a first step towards a fully autonomous micro quadrotor requiring no magnetometers, which will be able in the near future to "sense and avoid" obstacles in GPS-denied environments.

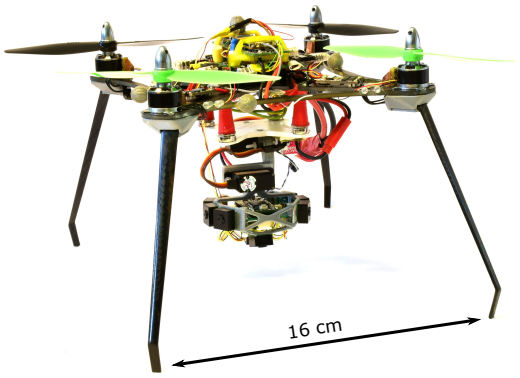


Figure 1: Photograph of the X4-MaG quadrotor with the gimbal case attached below (total mass: 395 g, span: 30 cm, autonomy: 6 min).

*Correspondance: julien.serres@univ-amu.fr; Tel.: +33(0)4 91 26 62 38

1 Introduction

Mimicking the flight of a tiny honeybee is still an arduous task [1, 2]. Since insect-sized micro air vehicles are increasingly becoming reality, however, [3, 4, 5, 6] it will be necessary to endow them in the future with sensors and flight control devices enabling them to perform all kinds of aerial maneuvers, including ground and obstacle avoidance, terrain-following and landing, in the same way as honeybees. This is a difficult challenge in the field of robotics, although drones can now be miniaturized thanks to recent advances in embedded electronics.

Stereovision-based strategies on board micro flying robots have recently provided means of exploring and avoiding obstacles indoors, but only under slow flight conditions [7, 8]. Micro LiDAR (Light Detection And Ranging devices) are also available for micro air vehicle applications (e.g., Centeye Inc. in the USA or LeddarTech Inc. in Canada), but these devices require a scanning system and have a high energy consumption. The optic flow (OF) is also a relevant visual cue which can be used for quickly sensing the configuration of the environment using either micro cameras or custom-made OF sensors.

Unfortunately, micro cameras sample tens of thousands of pixels with a refresh rate of only about 30 frames per second [9, 10], which is 10 times lower than the temporal resolution of the bees eye ($\sim$ 300 Hz, see [1]), and they also require much more computational resources than custom-made OF sensors. In addition, micro cameras are blinded by the changes in the light which occur when moving from one room to another, from indoor to outdoor conditions, or when encountering strong sunny to shadow contrasts. However, custom-made OF sensors lighten both the weight and the CPU load of micro flying robots , and enable them to make quasi panoramic OF measurements [11, 12, 13] as well as to work under various unpredictable lighting conditions. Because of their lack of visual stabilization, micro flying robots have to fly at low speeds despite the use of OF regulation principles [14, 10]. Up to now, no OF sensing strategies based on a stabilization system preventing the rotational effects have yet been implemented on board a micro flying robot.

In [9], the slope of the ground was assessed on board a quadrotor. In [15], 10 local OF measurements coupled to a least mean squares method were used to stabilize a minimalistic quasi-panoramic compound eye on the Beerotor robot with respect to the local downward slope, thus enabling the robot to avoid any very steep relief encountered. In [16], an

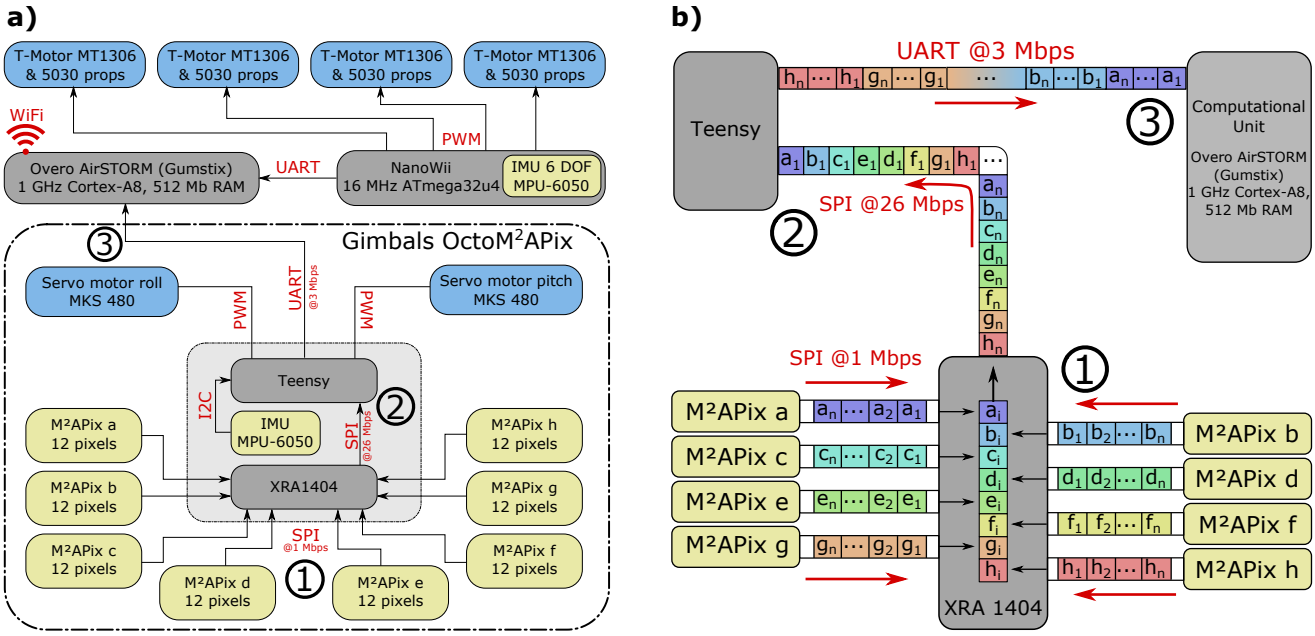


Figure 2: a) Component architecture of the X4-MaG quadrotor with the gimbal elements surrounded by a dash-dot line. The development boards and the data transmission units are shown in gray, the actuators in blue, and the sensors in yellow. b) Data diagram of how visual signals originating from the OctoM^{2}APix eye are processed: ① high speed synchronous data capture, ② data put in order, and ③ signal processing for optic flow computations.

innovative OF-based algorithm was tested in the horizontal plane as a means of measuring the robot's incidence angle when perceiving a slanting wall showing a moving texture. The results of our experiments showed that our minimalistic algorithm based on 20 local OF measurements could be used to determine the local visual heading with respect to the moving slanting wall with an error of less than 3° and a mean accuracy of 3° [16].

In the present paper, a bio-inspired micro flying robot, which will be able to perform both wall-following and ground-avoidance tasks in GPS-denied (Global Positioning System-denied) environments. This robot is fitted with a visual stabilization system involving 8 custom-made OF sensors which can respond appropriately in a 7-decade lighting range [17] and can measure the OF magnitude from 25 $^\circ/s$ to 1000 $^\circ/s$ [18]. It was established in real flight tests that this micro flying robot is able to estimate its incidence angle with respect to a slanting wall in order to restore its flight parallel to the wall. The results obtained in the present study show that our quasi-panoramic bio-inspired eye coupled to a computational unit can estimate this incidence angle in the 0° to 50° range when flying.

2 Micro flying robot

The micro flying robot presented here, a 320-gram quadrotor that can carry a maximum payload of 90 g, is based on the open-hardware X4-MaG model [19]. Thanks to an open-source Matlab/Simulink toolbox [20], it is possible to monitor this quadrotor in real time by wifi and change the relevant parameters directly during its flight. We have added to this quadrotor a custom-made 3D print gimbal case endowed with a magnet-based system of fixation facilitating its removal. This custom-made gimbal stabilizes the 8 OF sensors on the pitch and roll axes via two 10-gram serial servo motors (Fig. 1). The first group of 3 OF sensors oriented to the left measure a left OF, the second group of 3 sensors oriented to the right measure a right OF: each of these groups has a field of view of 92.8° in the horizontal plane and 13.4° in the vertical plane. The third group of two downward-oriented sensors measure the ventral OF with a field of view of 62.8° in the forward axis and 13.4° in the orthogonal axis. The gimbal system, which is stabilised in attitude during the robot's flight, has a total mass of 75 grams enabling the robot to fly for 6 minutes.

The quadrotor is fitted with a low-level autopilot based on the Nanowii (ATmega32u4, MultiWii) with a 6 degrees of freedom Inertial Measurement Unit (IMU) (Fig. 2a). This low-level control board makes it possible for the robot to be piloted manually and takes over from the Gumstix if failure of the latter occurs. All attitude and trajectory control processes based on the OF measurements are handled by the high-level autopilot based on an Overo AirSTORM Computer-On-Module (COM) (*Gumstix*) featuring a 1-GHz CPU DM3703 processor (*Texas Instruments*) comprising an ARM Cortex-A8 architecture. A Teensy 3.2 featuring a 72 MHz Cortex-M4 (PJRC) controls the gimbal servo motors and also reads

the 8 retinas (dedicated to OF measurements) and transmits the pixel data to the Overo AirSTORM (Fig. 2a, for details).

3 Optic flow sensors

The M^{2}APix sensor is a bio-inspired OF sensor based on a Michaelis-Menten Auto-adaptive Pixel analog silicon retina that can auto-adapt in a 7-decade lighting range and responds appropriately to stepwise changes of up to $\pm$ 3 decades [17]. Each M^{2}APix sensor is composed of 12 pixels distributed in two rows of six pixels offset by half the interpixel distance [17], and is able to make 10 local OF measurements with a high output refresh rate [18]. The local OF is measured using a time-of-travel algorithm based on a contrast detection matching method (called the thresholding method) running at only 1 kHz along the pixels rows axis.

The CPU load of the Overo AirSTORM is less than 3 percent per M^{2}APix sensor and the OF refresh rate is up to 99 Hz per M^{2}APix sensor with an OF ranging from 25 $^\circ/s$ to 1000 $^\circ/s$ [18].

4 A Gimbal system named OctoM^{2}APix

A custom-made PCB was designed for the gimbal system named OctoM^{2}APixto connect the following components (Fig. 2a): a Teensy 3.2 featuring a 72 MHz Cortex-M4 (PJRC), an 8-bit SPI GPIO EXPANDER called XRA1404 (Exar©), an Inertial Measurement Unit (IMU) MPU 6050, and connections to 8 M^{2}APix sensors and 2 servo motors belonging to the gimbal system, in order to stabilize the bio-inspired eye on the pitch and roll axes and thus remove any rotational OF components.

A M^{2}APix sensor works with a SPI bus running at a maximum rate of 1 MHz, which transmits a 256-bit data frame, and the maximum theoretical frame rate is therefore 3906 Hz. SPI devices communicate in the full duplex mode using a master-slave architecture with a single master. Only one SPI bus is usually available on dedicated electronic boards. The maximum frame rate possible to obtain data from the 8 slave M^{2}APix sensors is therefore less than 488 Hz, and the data will not be processed synchronously but sequentially (Fig. 2b). To give from the 8 M^{2}APix sensors higher frame rates and make the processing synchronous, a XRA1404 GPIO EXPANDER was used. The XRA1404 can read the logic state on 8 digital ports and send this data as a byte to an SPI bus working at a frequency of 26 MHz (Fig. 2b). Each of these 8 digital ports is used to read the bit to bit data frame of each M^{2}APix sensor (step ① in the Fig. 2b). As a result, the first bit in all the M^{2}APix data frames are arranged in one byte, which is sent on to the SPI bus. Each bit in each M^{2}APix sensor is processed in this way.

The Teensy electronic board reads the SPI bus at a frequency of 26 MHz from the XRA1404 and puts each data frame of each M^{2}APix sensor back into order (step ② in Fig. 2b). The M^{2}APix sensor sends a mean light value and 12 pixel values, each of which is coded in 10-bit values. The algorithm then selects only the pixel data. This leaves us with 12x10 bits per M^{2}APix sensor. Lastly, the Teensy removes the offset of the range used and deletes the last bit which is equivalent to the amplitude of the noise, to express the data in the 8-bit format. The 12 pixels coded with the 8-bit value of the 8 M^{2}APix sensors are then sent to the serial bus at a speed of 3 Mbps to a computational unit to compute the OF (step ③ shown in Fig. 2b).

5 Flying strategies

The OctoM^{2}APix is designed to make the X4-MaG drone both follow walls and adjust its height at relatively high speeds (up to 2.5 $m \cdot s^{-1}$) by removing the rotational OF component of the sensing performed by the 8 M^{2}APix sensors. Wall-following and ground-following behaviour will be obtained by merging OF regulation principles [21]. The aim of the flying strategies we propose to pursue in the near future will be to maintain the visual contact with the walls in order to make the robot fly in parallel along one of them. These strategies will involve adding another visual feedback loop controlling the yaw component by estimating the relative local angle between the nearest wall (e.g., the red rectangle in Fig. 4a) and the flying robot. Figure 3 and Eq. 1 explain how to obtain this incidence angle α from OF measurements with a very light solution in terms of the computational resources required, where c is cos, s is sin, and t is the tan function.

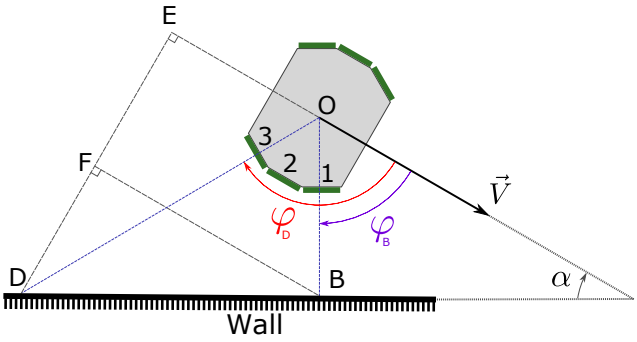


Figure 3: Geometry of the OctoM^{2}APix eye (centring in O) on the horizontal plane with a tapered wall on its left-hand side. Each green rectangle represents one M^{2}APix sensor. In the left-hand side, the M^{2}APix are numbered to make the experiments easier to understand.

$$\omega_B = \frac{V}{OB} \cdot s(\varphi_B) \qquad \omega_D = \frac{V}{OD} \cdot s(\varphi_D)$$

$$t(\alpha) = \frac{OD \cdot s(\varphi_D) - OB \cdot s(\varphi_B)}{OB \cdot c(\varphi_D) - OD \cdot c(\varphi_B)}$$

$$t(\alpha) = \frac{\frac{1}{\omega_D} \cdot s(\varphi_D)^2 - \frac{1}{\omega_B} \cdot s(\varphi_B)^2}{\frac{1}{\omega_B} \cdot s(\varphi_B) \cdot c(\varphi_B) - \frac{1}{\omega_D} \cdot s(\varphi_D) \cdot c(\varphi_D)}$$

$$\alpha = t^{-1}\left(\frac{\omega_B \cdot s(\varphi_D)^2 - \omega_D \cdot s(\varphi_B)^2}{\omega_D \cdot s(\varphi_B) \cdot c(\varphi_B) - \omega_B \cdot s(\varphi_D) \cdot c(\varphi_D)}\right) \quad (1)$$

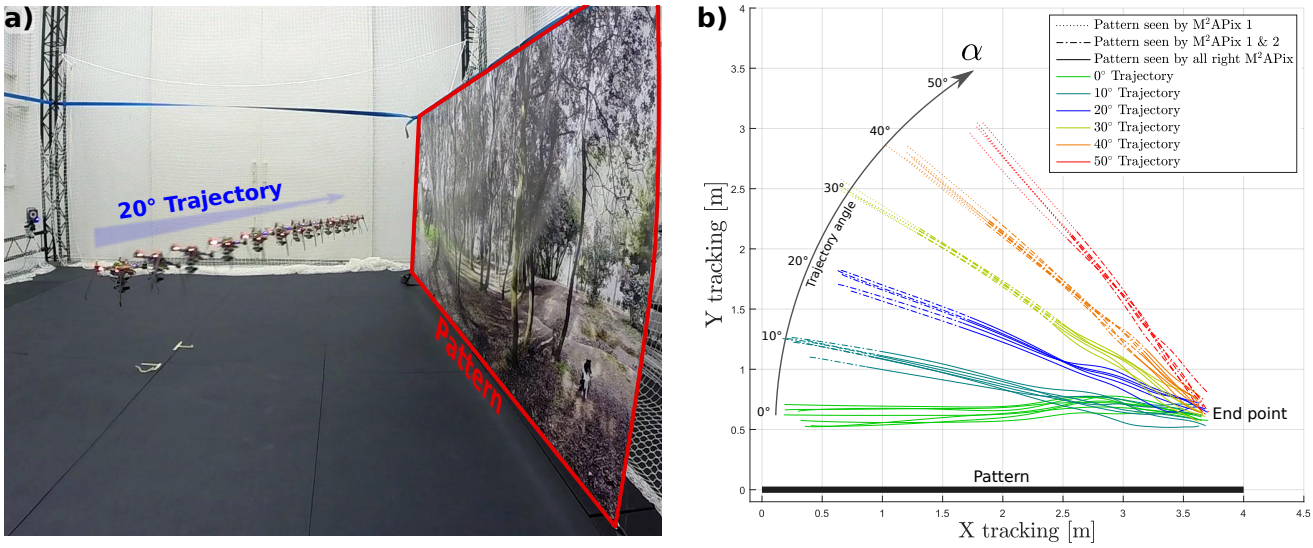


Figure 4: a) Chronophotograph of the X4-MaG drone at the Mediterranean Flying Arena. The flying robot followed a textured "wall" sloping at an angle of 20° for about 4.5 s (corresponding to one of the blue trajectories in b). b) Top view of all the trajectories taken in the Mediterranean Flying Arena[1]. 6 different trajectories were tested at the same height (0.8 $m \pm 0.01$ m) with a clockwise angle between the trajectories and the patterned wall. The X4-MaG drone followed each trajectory several times. Depending on the angle of the trajectory, either M^{2}APix #3 or both M^{2}APix #2 & #3 did not perceive the pattern on the wall due to their orientation (their positions are given by a dashed line and a dotted line, respectively).

6 Flying Experiment

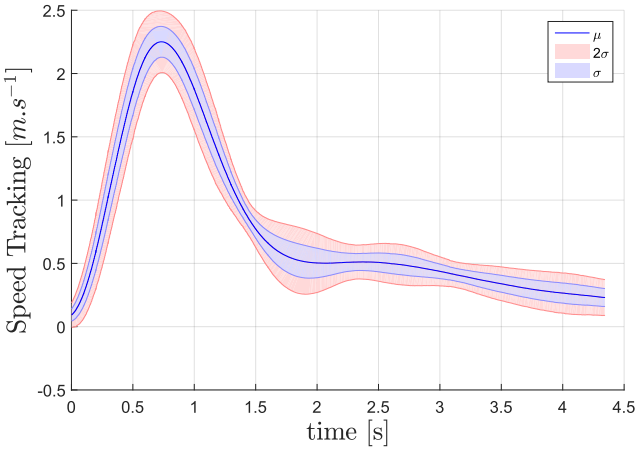


Figure 5: Speed profile (in XY plane) of the aerial robot for all the trajectories presented in Fig. 4b. μ is the mean value and σ is the standard deviation.

This flight experiment consisted in observing the dynamic responses of the OctoM^{2}APix sensor with all the components integrated during real flight. The X4-MaG flying at the Mediterranean Flying Arena[1] (6x8x6 m): its trajectory was controlled in the closed loop mode with a motion capture system (VICON^{TM}). A 4-m long wall covered with a natural pattern was hanging in the arena to generate OF when the X4-MaG approached it. The X4-MaG repeated various straight trajectories 6 times at the same height (0.8 $m \pm 0.01$ m) with the pattern on the right-hand side and with an angle of incidence α between its trajectory and the pattern ranging from 0° to 50° in 10° steps (Fig. 4). The three M^{2}APix sensors could therefore potentially detect the pattern and measure the OF during the experiments.

To test the case of future aggressive maneuvers in an indoor environment, the flights were then performed at high speed (up to 2.5 $m \cdot s^{-1}$, see Fig. 5) near obstacles (with a clearance of up to 0.5 m, see Fig. 4). The figure 5 gives the speed profiles of all the trajectories including one high acceleration (up to 6 $m \cdot s^{-2}$) phase and one high deceleration phase.

The figure 4b presents all the trajectories tracked. Due to the flying robots proximity with the patterned wall, aerodynamic perturbations were generated at the end of the trajectories, during the last metre (Fig. 4b). During a part of the trajectories from 10° to 50° shown by a dashed line, the pattern was not visible to the M^{2}APix #3 (oriented at an angle of 120° in Fig. 3) , and the dotted lines indicate the part of the trajectories where the pattern was not visible to the M^{2}APix #2 (oriented at an angle of 90° in Fig. 3). None of the OF measurements obtained during these dashed and dotted parts of the trajectories were included in the following statistical analyses.

A median filter was applied to the 10 local OF measurements obtained by each M^{2}APix sensor, and the M^{2}APix out-

[1] http://flying-arena.eu/

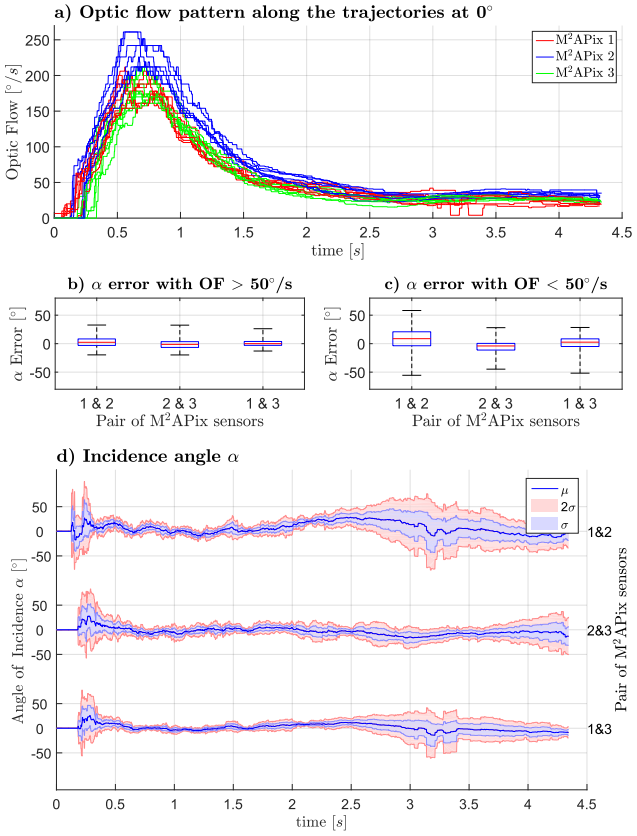


Figure 6: a) Optic flow measurements of M^{2}APix #1 (red), #2 (blue) & #3 (green), during all the trajectories performed at 0°. b) and c) Errors distribution of the angles of incidence α per pair of M^{2}APix sensors when all the optic flow measurements were greater than $50°/s$ in c), and less than $50°/s$ in d), where the red bar is the median, the black whiskers correspond to min and max values, and the blue box is the interquartile range (IQR) at 50 %. b) median ± IQR obtained for pair #1&#2 : $2.4° \pm 11.4°$; pair #2&#3 : $-1.2° \pm 10.1°$; pair #1&#3 : $0.0° \pm 6.8°$. c) median ± IQR for pair #1&#2 : $8.8° \pm 24.5°$; pair #2&#3 : $-4.0° \pm 11.7°$; pair #1&#3 : $2.6° \pm 13.4°$. d) Angle of incidence α computed for each pair of M^{2}APix sensors. μ is the mean value and σ is the standard deviation.

puts were then used in pairs to compute Eq. 1 in order to estimate the local angle of incidence α (Fig. 3). We thus obtained 3 measurements of the α angle from these pairs (#1&#2, #2&#3 and #1&#3). Statistical results of the angular measurements obtained from each set of trajectories (from 0° to 50°) are presented in Figs. 6 to 11 in 10° steps.

The optical parameters of the M^{2}APix sensors were chosen so as to be able to measure OF values greater than 50° /s (up to $1000°/s$, see [18]) accurately. The angle of incidence α was therefore estimated more accurately with OF values greater than 50° /s, as shown in Figs. 6b-11b than in Figs 6c-11c. The lowest α errors were consistently obtained with the pair of M^{2}APix sensors #1 and #2 (Figs. 6b,c-11b,c), where

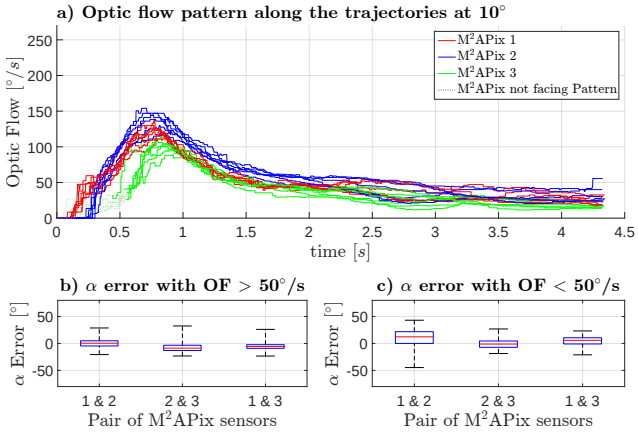


Figure 7: a) Optic flow measurements made by M^{2}APix #1 (red), #2 (blue) & #3 (green) during all the trajectories performed at 10°. b) c) Errors distribution of the angles of incidence α recorded per pair of M^{2}APix when all the optic flow measurements were greater than $50°/s$ in b) and less than $50°/s$ in c), where the red bar is the median, the whiskers correspond to min and max values and the blue box is the IQR at 50 %. OF measurements plotted in dotted lines were not generated by the pattern on the wall and were therefore not included in the results presented in b) and c) in this figure and the next ones. b) median ± IQR for pair #1&#2 : $0.5° \pm 9.6°$; pair #2&#3 : $-8.7° \pm 9.8°$; for pair #1&#3 : $-5.6° \pm 6.7°$. c) median ± IQR for pair #1&#2 : $12.3° \pm 21.8°$; pair #2&#3 : $-1.2° \pm 11.7°$; pair #1&#3 : $5.7° \pm 11.5°$.

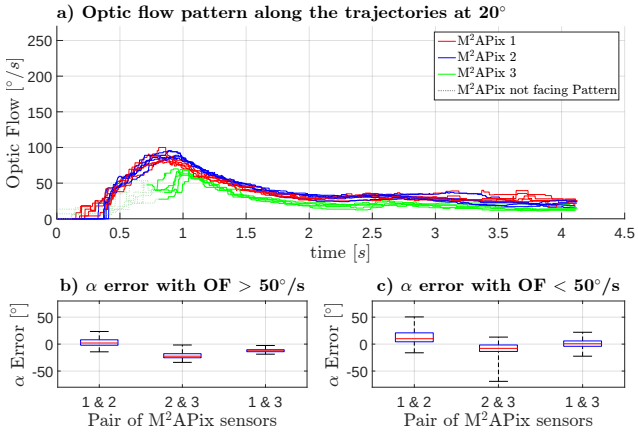


Figure 8: Results obtained on the trajectories at 20°. b) median ± IQR for pair #1&#2 : $2.0° \pm 10.1°$; for pair #2&#3 : $-22.4° \pm 7.6°$; pair #1&#3 : $-12.5° \pm 3.7°$. c) median ± IQR for pair #1&#2 : $10° \pm 16.6°$; pair #2&#3 : $-7.9° \pm 12.0°$; pair #1&#3 : $0.6° \pm 9.9°$.

the median error was less than 2.5°. This pair of sensors was that which was the most oriented towards the frontal part of the OF field.

The visual strategy developed in this study should therefore be consistent with the confidence range of the angle of incidence α estimated when the OF values measured on the

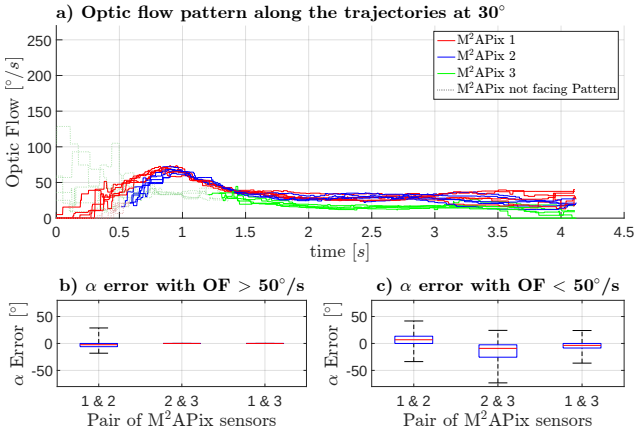


Figure 9: Results obtained on the trajectories 30°. The M²APix 3 did not measure optic flows greater than $50°/s$, which explains the lack of results on the pair of M²APix #2&#3 and #1&#3 in b). b) median ± IQR for pair #1&#2 : $-2.3° \pm 6.0°$. c) median ± IQR for pair #1&#2 : $6.9° \pm 13.4°$; pair #2&#3 : $-9.1° \pm 23.0°$; pair #1&#3 : $-3.5° \pm 8.5°$.

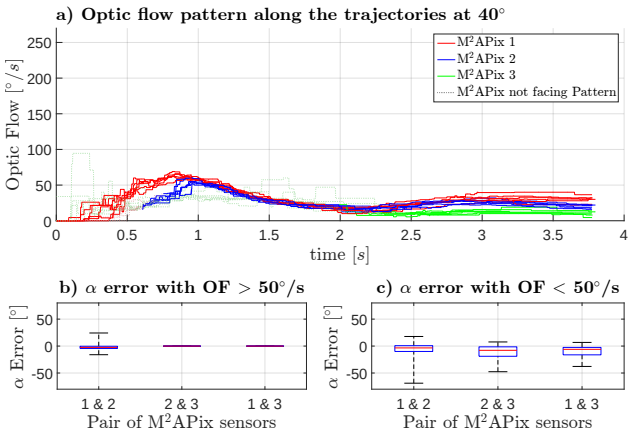


Figure 10: Results obtained on the trajectories 40°. b) median ± IQR for pair #1&#2 : $-2.5° \pm 4.1°$. c) median ± IQR for pair #1&#2 : $-3.4° \pm 10.7°$; pair #2&#3 : $-7.9° \pm 17.5°$; pair #1&#3 : $-6.0° \pm 13.8°$.

micro aerial vehicle were high, i.e. at high speeds, as well as when the micro aerial vehicle was travelling very near the walls.

7 Conclusion

The 395-gram X4-MaG quadrotor fitted with a gimbal system and a set of 8 custom-made optic flow sensors provides an appropriate flying platform for testing optic-flow regulation principles during real flight with a view to mimicking honeybees' flight performances. The gimbal system makes it possible to minimize the effects of the rotational component of the optic flow measured by the quasi-panoramic bio-inspired eye. The results obtained in these experiments show that our optic flow based algorithm can estimate the drone's local angle of incidence with respect to a slanting wall in the 0° to 50° range during flight with an error of less than 2.5°, using the optic flow generated on either side (left or right). To measure larger angles of incidence (from 50° to 90°), the fronto-bilateral optic flow could be used in the similar way to that simulated in [22], but this possibility is beyond the scope of the present study and would require measuring the low optic flow values occurring close to the focus of expansion.

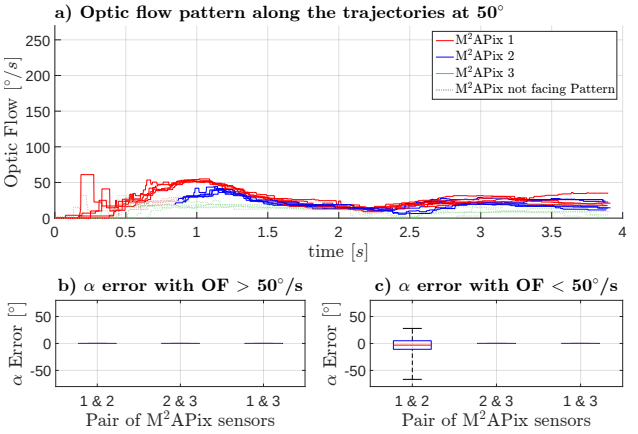


Figure 11: Results obtained on the trajectories at 50°. The pattern on the wall was not visible to M²APix #3 during all these trajectories, which explains the lack of data on the pair of M²APix #2&#3 and #1&#3 in c). c) median ± IQR for pair #1&#2 : $-3.1° \pm 15.8°$.

In the near future, flying at high speed (up to 1 m/s) near obstacles (at a distance of less than 0.5 m) may be possible thanks to the large range of optic flow measurements which can now be made, namely up to $1000°/s$. The advantages of the custom-made optic flow sensors used here include low power consumption, low computational requirements, and robustness to high dynamic range lighting conditions (7 decades). In conclusion, the present X4-MaG drone fitted with smart visual sensors is the first step towards designing airborne vehicles capable of autonomous navigation requiring no magnetometers in GPS-denied environments.

Acknowledgements

We thank J. Diperi for the mechanical design of the flying robot and the test bench, M. Boyron for his involvement in the overall electronic design of the flying robot, and K. Contamain, S. Mafrica, and F. Colonnier for their help with both the electronic development and the programming of the M²APix sensors. This research was supported by the French Direction Générale de l'Armement (DGA), CNRS, Aix-Marseille Université, the Provence-Alpes-Côte d'Azur region and the French National Research Agency for Research (ANR) in the framework of the Equipex/Robotex project.

REFERENCES

[1] M. V. Srinivasan, "Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics," *Physiological reviews*, vol. 91, no. 2, pp. 413–460, 2011.

[2] J. R. Serres and F. Ruffier, "Optic flow-based collision-free strategies: From insects to robots," *Arthropod Structure & Development*, 2017 (in press).

[3] P.-E. J. Duhamel, N. O. Pérez-Arancibia, G. L. Barrows, and R. J. Wood, "Altitude feedback control of a flapping-wing microrobot using an on-board biologically inspired optical flow sensor," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 4228–4235, IEEE, 2012.

[4] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.

[5] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood, "Controlled flight of a biologically inspired, insect-scale robot," *Science*, vol. 340, no. 6132, pp. 603–607, 2013.

[6] O. Dunkley, J. Engel, J. Sturm, and D. Cremers, "Visual-inertial navigation for a camera-equipped 25g nano-quadrotor," in *IROS2014 aerial open source robotics workshop*, 2014.

[7] K. McGuire, G. de Croon, C. De Wagter, K. Tuyls, and H. Kappen, "Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1070–1076, 2017.

[8] Centeye, *Centeye nano unmanned aircraft system with 360-degree stereo vision*. Washington, DC: http://www.centeye.com/small-nano-uas-autonomy, 2016.

[9] G. De Croon, H. Ho, C. De Wagter, E. Van Kampen, B. Remes, and Q. Chu, "Optic-flow based slope estimation for autonomous landing," *International Journal of Micro Air Vehicles*, vol. 5, no. 4, pp. 287–297, 2013.

[10] C. Sabo, E. Yavuz, A. Cope, K. Gumey, E. Vasilaki, T. Nowotny, and J. A. Marshall, "An inexpensive flying robot design for embodied robotics research," in *Neural Networks (IJCNN), 2017 International Joint Conference on*, pp. 4171–4178, IEEE, 2017.

[11] J. Keshavan, G. Gremillion, H. Alvarez-Escobar, and J. S. Humbert, "Autonomous vision-based navigation of a quadrotor in corridor-like environments," *International Journal of Micro Air Vehicles*, vol. 7, no. 2, pp. 111–123, 2015.

[12] R. J. Moore, K. Dantu, G. L. Barrows, and R. Nagpal, "Autonomous mav guidance with a lightweight omnidirectional vision sensor," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3856–3861, IEEE, 2014.

[13] A. Briod, J.-C. Zufferey, and D. Floreano, "A method for ego-motion estimation in micro-hovering platforms flying in very cluttered environments," *Autonomous Robots*, vol. 40, no. 5, pp. 789–803, 2016.

[14] C. Sabo, A. Cope, K. Gurny, E. Vasilaki, and J. A. R. Marshall, "Bio-Inspired Visual Navigation for a Quadcopter using Optic Flow," *AIAA Infotech @ Aerospace*, no. January, pp. AIAA 2016–0404, 2016.

[15] F. Expert and F. Ruffier, "Flying over uneven moving terrain based on optic-flow cues without any need for reference frames or accelerometers," *Bioinspiration and Biomimetics*, vol. 10, 2015.

[16] E. Vanhoutte, F. Ruffier, and J. Serres, "A quasi-panoramic bio-inspired eye for flying parallel to walls," in *Sensors, 2017 IEEE*, (paper 1332, Glasgow, Scotland, UK), IEEE, 2017 (in press).

[17] S. Mafrica, S. Godiot, M. Menouni, M. Boyron, F. Expert, R. Juston, N. Marchand, F. Ruffier, and S. Viollet, "A bio-inspired analog silicon retina with Michaelis-Menten auto-adaptive pixels sensitive to small and large changes in light," *Optics Express*, vol. 23, no. 5, p. 5614, 2015.

[18] E. Vanhoutte, S. Mafrica, F. Ruffier, R. Bootsma, and J. Serres, "Time-of-Travel Methods for Measuring Optical Flow on Board a Micro Flying Robot," *Sensors*, vol. 17, no. 3, p. 571, 2017.

[19] A. Manecy, N. Marchand, F. Ruffier, and S. Viollet, "X4-MaG : A Low-Cost Open-Source Micro-Quadrotor and Its Linux-Based Controller," *International Journal of Micro Air Vehicles*, vol. 7, no. 2, pp. 89–110, 2015.

[20] A. Manecy, N. Marchand, and S. Viollet, "RT-MaG: An open-source SIMULINK toolbox for Linux-based real-time robotic applications, year=2014, pages=173-180, doi=10.1109/ROBIO.2014.7090326, month=Dec,"

[21] G. Portelli, J. Serres, F. Ruffier, and N. Franceschini, "Modelling honeybee visual guidance in a 3-D environment," *Journal of Physiology Paris*, vol. 104, no. 1-2, pp. 27–39, 2010.

[22] J. R. Serres and F. Ruffier, "Biomimetic autopilot based on minimalistic motion vision for navigating along corridors comprising u-shaped and s-shaped turns," *Journal of Bionic Engineering*, vol. 12, no. 1, pp. 47–60, 2015.

Loosely Coupled Stereo Inertial Odometry on Low-cost System

HaoChih, LIN[*] Francois, Defay[†]

Abstract — We present a fast and robust stereo visual inertial odometry system which is friendly for low cost sensor and single board computer (SBC). Comparing against other research which uses tightly coupled algorithms or nonlinear optimization to increase accuracy in custom powerful hardware or desktop environment, our system adopts the loosely coupled ESKF to limit the computational complexity in order to fit limited CPU resources and run in real-time on an ARM based SBC. The experiments demonstrates our method could be implemented in both indoor and outdoor scenarios with competitive accuracy. Furthermore, the usage of forward facing stereo cameras also provides the ability of obstacles avoidance. The result are released as an open sourced Robot Operation System (ROS[1]) package.

I Introduction

In order to make a micro aerial vehicle (MAVs) achieve fully autonomous navigation in GPS denied environment, the one of fundamental challenge is to obtain fast, robust and accurate 3D egomotion estimation with others sensors. Recently, the visual inertial odomerty (VIO) attracts significant attentions in MAV researching field because the algorithm efficiently integrates the rich representation of a scene captured in an image, together with the accurate short-term measurements by IMU. In addition, comparing to heavy and expensive 3D laser scanner or rgbd sensor, the VIO approach could be able to present similar performance with much less value in both price and size aspects which may increase the endurance and maneuverability of MAVs.

In this paper, we propose a loosely coupled stereo inertial odometry based on error state kalman filter (ESKF) algorithm. The main contribution of our method is to limit the computational loading by using the characteristics of loosely coupled architecture which uses the fixed dimension of state space in order to be implemented on low-cost ARM based SBC (Odroid XU4[2]). The algorithm could run at around 100Hz depending on IMU publishing rate and works with forward-facing stereo cameras, allowing for fast flight and removing the need for a second camera for collision avoidance. Apart from this, our approach provides steady metric scale information from disparity matching to meet the power on and go requirement without requiring specific initialization.

[*] Advanced Robotics Co., Ltd, Taiwan. Email: haochihlin93@gmail.com
[†] ISAE-Supaero, France. Email: francois.defay@isae-supaero.fr
[1] http://wiki.ros.org/
[2] http://www.hardkernel.com/main/main.php

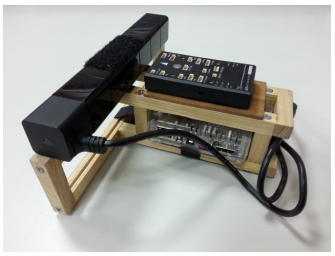


Figure 1: The hardware overview of proposed VIO system composed of ARM based SBC, low cost IMU, and modified PS4-Eye.

II Related Work

The fusion of Inertial Measurement Units (IMU) with a visual sensors has become popular in robotics community. Recent work can be categorized to two approaches: tightly coupled and loosely coupled system. The former, e.g. [1–3], jointly estimates the image features information with IMU data. The latter, [4–6], take the visual odometry(VO) as a independent black box. By means of the characteristics of loosely coupled architecture, we could divide our algorithm into two main sections: VO and ESKF part.

II.1 VO part

There is a vast amount of existing visual algorithms to estimate the locomotion from traditional feature extraction and matching method, for instance PTAM [7] and ORB-SLAM [8], to semi dense approaches, which uses features to locate small patches, then operates directly on pixel intensities. The representative of this method is SVO [9] and LSD-SLAM [10]. In addition, thanks to the significant improvement on CPU/GPU computing power, the current state-of-art method avoid feature detection process instead using all pixel information directly from image stream. The famous achievements are [11, 12]. However, these two approaches generally rely on powerful or custom hardware. Hence, this is the main reason that we choose the traditional features based method for our stereo visual odometry developing.

II.2 ESKF part

The one common way to fuse IMU with odometry or other types of sensor data is EKF (Extended Kalman Filter). Although, recently, some researches adopts other nonlinear optimizer to get more accuracy and robust performance, the computing cost of the method is too larger to be implemented on ARM-based SBC. As a result, in our system, we apply

ESKF algorithm to overcome the limited CPU issue and also the main drawbacks of traditional EKF architecture. According to the experiments and explanation from [13, 14], ESKF has following remarkable assets: the orientation error-state is minimal to avoid over-parametrization issue, the error-state system is always operating close to the origin to guarantee the linearization validity and the error dynamics are slow, hence the rate of correction could be lower than prediction rate.

The rest of the paper is organized as follows. Section III shows the definition of the notations. Section IV describes the core algorithm of VO and statical way to remove outliers. The architecture of ESKF is fully explained in section V, followed by section VI, which describes the detail of practical implementation. Finally, section VII presents the experimental result and concludes the paper as well as future research directions.

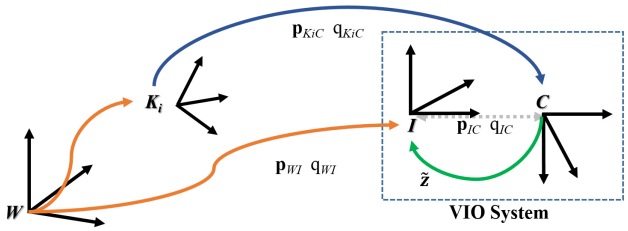


Figure 2: Visualization of the different coordinate frames. Blue lines denote the transformation obtained by VO, red lines stand for the prediction part of ESKF and green lines depict the update part

III Notation and Definitions

We employ the following notation throughout this work and are illustrated in Figure 2. For the Frame definitions: W is the world frame, F_i is the i-th keyframe frame, I is IMU frame and C is camera frame. A translation vector from frame A to frame B expressed in frame A is denoted as bold $\mathbf{p}_{ab}$. A rotation matrix from A to B is expressed as $\mathbf{R}_b^a \in SO3$ which can also be denoted as a quaternion $\bar{q}_{ab}$. In this paper, we adhere the Hamilton convention [15], which is right-handed and widespread used in robotics, to define a quaternion by $q = [q_w + q_x + q_y + q_z] = [q_w, \mathbf{q}]$. We adopt the notation introduced in [1] to handle the quaternion multiplication as $q_{ac} = q_{ab} \otimes q_{bc}$.

IV Stereo Visual Odometry

Take the limits of CPU resources into account, our VO approach adopt ORB [16] for both feature detection and description instead of robust but slow descriptors like SURF or SIFT. We also apply the concept of keyframe to decrease the drift effect. In the other hand, unlike other VO researches [8, 17] emphasis on back-end optimization process, e.g. bundle adjustment or graphic based loop closure, to tackle the

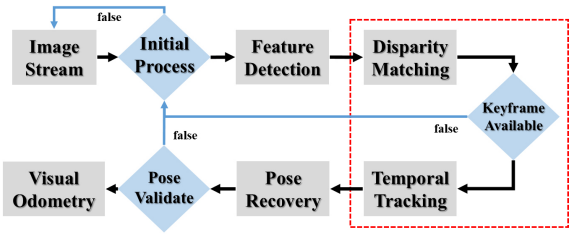


Figure 3: The system overview of proposed VO algorithm.

long-term global localization issue, we are more focusing on the short-term accuracy and robustness because the VO outcome will be used as the local correction inputs for the ESKF based on the previous keyframe pose rather than global correction referred to the initial VO frame. Especially speaking, the continuity of VO does not hugely affect the system performance, hence the back-end optimization process is not mandatory. This decision also descends the demands of CPU resource.

IV.1 Disparity Computing

The system overview of our VO approach is illustrated in Figure 3. Starting from disparity block, the scale metric information can be calculated immediately and continuously from the initial phase due to known camera calibration matrix. The matching method in this block is mainly depends on the brutal force matcher with additional epipolar constraints. Then the succeeded matching pairs will be sent into tracking block. The tracking part is composed of two procedures: the Temporal Features Matching and Pose Recovery.

IV.2 Temporal Features Matching

The state-of-art method to handle the issue is using the circular searching pattern strategy, e.g. [1, 8], which searches similar features pairs from both temporal and disparity correspondences to reduce the outlier probability, however, since the stereo camera in our system is hardware synchronizes and its baseline is very short, the effect is not significant compared to other system configuration. Therefore, our system only relies on left side images to search correspondences between two sets of descriptors. In order to minimize the dissimilarity score between each pair of descriptors p_{ij} and increase the matching robustness, we adopt exhaustive searching strategy, also called brutal force, to find all possible pairs. The main defect of this strategy is the bad correspondences, especially for indoor environments with many repetitive patterns. To solve this issue, we introduced a statistics model to determine which pair is outlier. Every time when the exhaustive searching strategy generates a list of matching pairs, the system will compute the mean μ and standard deviation σ based on Hamming distance ($HD(p_{ij})$) of ORB descriptors. Then the system would decide the outlier by the following equation:

$$if\ HD(p_{ij}) > \mu + a\sigma, \quad then\ p_{ij} \in Outlier \qquad (1)$$

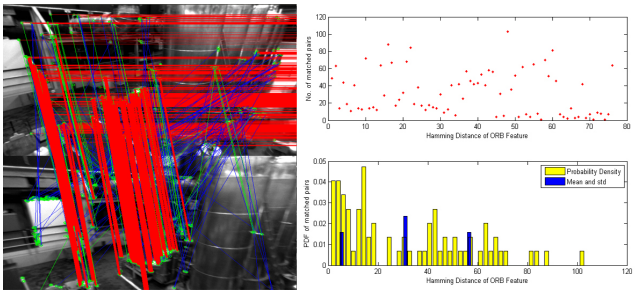


Figure 4: Left: temporal image pair for features tracking matches. Right: histogram to show distribution of matched pair Hamming distance.

where a is a custom defined weighted factor. The outcome of this approach is quite competitive to the circular strategy. Figure 4 demonstrates the performance of our strategy from both raw images and histogram graphic views. The green lines of raw images stands for original brutal force matching result. The blue lines represents the remaining pairs after statistics based outliers removing. The red line is the inlier of PnP RANSAC outputs, which will be explained in the following section.

IV.3 Pose Recovery

There are three types of standard approaches for motion estimation: 2D-2D, 2D-3D and 3D-3D. The first method is based on specified 2D features correspondences in a image pair. The popular related algorithms are 5 and 8 points solution [18]. The main issue of 2D-2D is observability of scale information. The final method also called the point cloud registration which is widely implemented on RGBD or 3D laser scanner since the depth data is highly reliable and accuracy. The popular algorithms include ICP [19] and Bundle Adjustment [20]. None of both methods satisfy our scenario, thus, we choose the 2D-3D approach in our VO system. Currently, there are 3 existing algorithms supported by OpenCV[3] library, they are: Iterative PnP, EPNP and P3P [21, 22]. The main concept of the algorithm is to find the minimal reprojected error by Equation 2 from 3D structure and 2D image correspondences.

$$T_{k-1}^{k} = \underset{T_{k-1}^{k}}{\arg\min} \sum_{i} \left\| q_k^i - \hat{q}_{k-1}^i \right\|^2 \tag{2}$$

where T_{k-1}^{k} is a transformation matrix from k-1 to k, q_k^i is a 2D description vector of i-th feature in k frame. $\hat{q}_{k-1}^i$ is a projected corresponding 2D descriptor from k-1 matched 3D point.

Because of various characteristics among these options, we let the users to select the one satisfied their demands. In addition, to tackle the bad corresponding matching pair generated by previous block, all three algorithms are accompanied by RANSAC process in order to remove outliers.

[3] http://opencv.org/

IV.4 Keyframe Selection and Initialization

The accumulated drift issue is the most common problem among different types of odometry. For the VO system, the most efficient way to decrease the rate of drift accumulation is to introduce the keyframe concept like [1–3, 7–12]. Therefore, we apply the keyframe architecture into our system. The first keyframe will be selected during the initialization process. The VO system will continuously re-initialize until the first keyframe is selected successfully based on the quality of disparity and tracking matching. When the re-initialization occurs, which will be triggered as system detects the lose tracking phenomenon, system will drop current keyframe and try to set up a new one.

IV.5 Practical Aspect

Although our system allows the users to modify parameters to meet their requirements, to get the optimal performance on limited CPU resources SBC, we suggest the resolution of input video stream should not be higher than 320*240. The maximum number of extracted feature points from an image extremely affects the realtime performance.

Moreover, as mentioned above, the primary purpose of our VO system is to provide reliable and accuracy short-term odometry based on the current keyframe pose. Hence, we raise the threshold for tracking matching constraints and re-initialize whole VO system right away when encountering lose tracking scenario. The outcomes largely decrease the CPU loading. The format of output data is a transformation matrix T_{KC}, consists of q_{K_iC} and $\mathbf{p}_{K_iC}$, from latest Keyframe (K_i) to Camera frame (C) expressed in the Keyframe.

V Error State Kalman Filter

The most well-known papers to illustrate the detail of ESKF formula derivation are [13, 14]. This section mainly adheres the concepts shown in both papers, but we replace the JPL's convention with Hamilton's for quaternion definition, in addition, with the benefits of loosely coupled system property, the size of state space and related matrix dimension are all fixed.

In the following subsections, the state of the filter is denoted as a 16-elements state vector $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}_{WI} & \mathbf{v}_{WI} & q_{WI} & \mathbf{b}_a & \mathbf{b}_\omega \end{bmatrix}^T \tag{3}$$

where $\mathbf{p}_{WI}$ is the position of IMU frame's origin (I) in the inertial world frame (W). $\mathbf{v}_{WI}$ is a velocity vector of IMU. q_{WI} describes a rotation from IMU frame to world frame. $\mathbf{b}_a$ and $\mathbf{b}_\omega$ are bias vector of gyro and acceleration respectively. The measurements of the IMU are known to be subject to different error terms, such as a process noise and a bias. Thus, for the real angular velocities $\boldsymbol{\omega}$ and the real accelerations $\mathbf{a}$,

which is expressed in sensor frame, we have following relations:

$$\boldsymbol{\omega} = \boldsymbol{\omega}_m - \mathbf{b}_\omega - \mathbf{n}_\omega, \quad \mathbf{a} = \mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a \tag{4}$$

where the subscript m denotes the measured value. $\mathbf{n}_a$ and $\mathbf{n}_\omega$ are zero mean white Gaussian noise processes. The bias is non-static and simulated as a random walk process: $\dot{\mathbf{b}}_\omega = \mathbf{n}_\omega$, $\dot{\mathbf{b}}_a = \mathbf{n}_a$.

In order to make the filter more robust and converges faster, we assume the calibrated transformation between IMU frame (I) and Camera frame (C) are fixed. The calibration states are denoted as q_{IC} for rotation from camera to IMU frame and $\mathbf{p}_{IC}$ for translation of camera frame expressed in IMU frame.

Because of limited space and considering the practical implementation issue of numerical integration on embedded system, the following subsections only depicts formulas in discrete time domain. For more information of continuous time model, the reader is referred to [13, 14].

V.1 Prediction

The main concept of the error-state filter is to treat the true state ($\mathbf{x}_t$) as a composition of the nominal state ($\mathbf{x}$: the state is integrated by high-frequency IMU data without considering noise terms and other possible model imperfections) and the error state ($\delta\mathbf{x}$: the state is in charge of collecting all the noise and perturbations). The relation is expressed as: $\mathbf{x}_t = \mathbf{x} \oplus \delta\mathbf{x}$, where the operator $\oplus$ indicates a generic composition. For all vectors in state space are equal to the typical addition symbol (+), except for the quaternion, it is equivalent to the quaternion multiplication symbol ($\otimes$) Additionally, ESKF also continuously predicts a Gaussian estimate of the error-state.

The following differential equations govern the nominal state kinematics:

$$\mathbf{p}_{k+1} \leftarrow \mathbf{p} + \mathbf{v}\Delta t + 0.5(\Omega_{(q)}(\mathbf{a}_m - \mathbf{a}_b) + \mathbf{g})\Delta t^2 \tag{5}$$

$$\mathbf{v}_{k+1} \leftarrow \mathbf{v} + (\Omega_{(q)}(\mathbf{a}_m - \mathbf{a}_b) + \mathbf{g})\Delta t \tag{6}$$

$$q_{k+1} \leftarrow q \otimes q[(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\Delta t] \tag{7}$$

$$\mathbf{a}_b, \boldsymbol{\omega}_b, \mathbf{g} \quad are\ constant \tag{8}$$

where $\Omega_{(q)}$ is the quaternion multiplication matrix of nominal quaternion q and $q[\boldsymbol{\omega}]$ is a quaternion converted by a rotation vector $\boldsymbol{\omega}$. Becasue the rate of prediction depends on IMU publishing rate which is usually higher than 100 Hz and almost constant, we assume the angular velocity over the period Δt is also invariant. Therefore, $q[\boldsymbol{\omega}]$ could be interpreted as the Taylor series of $\boldsymbol{\omega}\Delta t$ by midward zeroth order integration:

$$q_{k+1} \leftarrow q \otimes (1 + \frac{1}{2}\bar{\boldsymbol{\omega}}\Delta t + \frac{1}{2!}(\bar{\boldsymbol{\omega}}\Delta t)^2 + ...) \tag{9}$$

where $\bar{\boldsymbol{\omega}}_k = 0.5(\boldsymbol{\omega}_{k+1} + \boldsymbol{\omega}_k)$. In real-time implementation, the system calculates the Taylor series up to 4th order.

To increases numerical stability and handles the minimal representation of quaternion computing, we defined the error quaternions as $\delta q = q \otimes \hat{q} \approx [1, \frac{1}{2}\delta\boldsymbol{\theta}^T]^T$. Thus, we define a 15-elements error state vector:

$$\delta\mathbf{x} = \begin{bmatrix}\delta\mathbf{p} & \delta\mathbf{v} & \delta\boldsymbol{\theta} & \delta\mathbf{b}_a & \delta\mathbf{b}_\omega\end{bmatrix}^T \tag{10}$$

There are several ways to illustrate the compact form of the error-state kinematics, we modify the form listed in [5] which is expressed as following:

$$\delta\mathbf{x} \leftarrow \mathbf{F}_d(\mathbf{x}, \mathbf{u_m})\delta\mathbf{x} + \mathbf{G}_c\mathbf{Q}_d\mathbf{G}_c^T\mathbf{n} \tag{11}$$

where $\mathbf{u}_m = [\mathbf{b}_a\ \mathbf{b}_\omega]^T$ is the input vector, and $\mathbf{n} = [\mathbf{n}_a\ \mathbf{n}_\omega]^T$ is the noise vector ($\mathbf{n} \sim N\{0, \mathbf{Q}_d\}$). $\mathbf{F}_d \in \mathbb{R}^{15\text{x}15}$ and $\mathbf{Q}_d = diag(\sigma^2_{\boldsymbol{b}_a}\Delta t^2, \sigma^2_{\boldsymbol{b}_\omega}\Delta t^2, \sigma^2_{\boldsymbol{n}_a}\Delta t, \sigma^2_{\boldsymbol{n}_\omega}\Delta t)$ are the discrete state transition matrix and noise covariance matrix respectively. $\mathbf{G}_c = [zeros(3, 12);\ identity(12)]$. To find the ESKF prediction equation, we calculate the expectation of Equation (11) and its state covariance matrix $\mathbf{P}$:

$$\hat{\delta\mathbf{x}} \leftarrow \mathbf{F}_d(\mathbf{x}, \mathbf{u_m})\hat{\delta\mathbf{x}} \tag{12}$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}_d\mathbf{P}_{k|k}\mathbf{F}_d^T + \mathbf{G}_c\mathbf{Q}_d\mathbf{G}_c^T \tag{13}$$

Generally, the error-state $\delta\mathbf{x}$ is set to zero during initialization phase, so Equation (12) always returns zero. For the structure of matrix $\mathbf{F}_d$, we adopt the compact form defined in [realtime metric], with the small angle approximation when $|\boldsymbol{\omega}| \rightarrow \mathbf{0}$. We now proceed the ESKF prediction as follows:

1. Propagate the nominal state variables according to Equation from (4) to (9)
2. Calculate the matrix $\mathbf{F}_d$ and $\mathbf{Q}_d$
3. Compute the state covariance matrix by Equation (13)

V.2 Updates

The update procedure will be triggered whenever a validate visual measurement is generated by the stereo odometry algorithm. As described above, the VO system adopts the keyframe concept to reduce the effect of accumulated drift, hence, the odometry used in update phase represents the transformation from the state at keyframe to the present. Unlike other loosely coupled approaches, e.g. [6], used globe pose of visual measurements to update EKF states, we choose the local odometry to correct the ESKF error states. The main advantage of this approach is the computational delay or losing track scenario will not hugely affect the stability and performance of ESKF.

The measurement model in our case is quite straightforward since the transformation between IMU frame and Camera frame is assumed to be constant and known by means of the system extrinsic calibration. The measurement vector ($\mathbf{z}^{vo}$) is expressed as follows:

$$\mathbf{z}^{vo} = \begin{bmatrix}\mathbf{p}^{vo}_{WI} \\ q^{vo}_{WI}\end{bmatrix} = \begin{bmatrix}\Omega_{(q_{WKi})}(\Omega_{(q_{CI})}\mathbf{p}_{KiC} + \mathbf{p}_{CI}) + \mathbf{p}_{WK_i} + \mathbf{n}_p \\ q_{WK_i} \otimes (q_{CI} \otimes q_{K_iC} \otimes q_{CI}^{-1}) \otimes \delta q_{n_q}\end{bmatrix} \tag{14}$$

where $\mathbf{p}_{WI}^{vo}$ represents the translation from world to imu frame observed by VO system, q_{WI}^{vo} is for rotation. $\mathbf{n}_p$ and $\mathbf{n}_q$ are zero mean, white Gaussian noise of the visual measurement. q_{K_iC} and $\mathbf{p}_{K_iC}$ are the odometry output generated by VO algorithm.

Then we calculate the update residual between propagated state and measurement state:

$$\tilde{\mathbf{z}} = \mathbf{z} \ominus \hat{\mathbf{z}} = \begin{bmatrix} \mathbf{p}_{WI}^{vo} - \mathbf{p}_{WI}^{imu} \\ q_{WI}^{vo} \otimes q_{WI}^{imu\,-1} \end{bmatrix} \tag{15}$$

Then we approximates the $q_{WI}^{vo} \otimes q_{WI}^{imu\,-1} \approx [1, -0.5\tilde{\boldsymbol{\theta}}^T]^T$, and linearise the Equation (15) with respect to the error-state:

$$\tilde{\mathbf{z}} = \mathbf{H}\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{0}_{3x6} & \mathbf{0}_{3x6} \\ \mathbf{0}_{3x6} & \mathbf{I}_{3x3} & \mathbf{0}_{3x6} \end{bmatrix} \tilde{\mathbf{x}} \tag{16}$$

where $\tilde{\mathbf{x}} \in \mathbb{R}^{15\mathrm{x}1}$. Now we can apply the standard EKF update procedure:

1. Compute the residual from Equation (15)
2. Compute the innovation matrix: $\mathbf{S} = \mathbf{HPH}^T + \mathbf{V}$
3. Compute the Kalman gain: $\mathbf{K} = \mathbf{PH}^T\mathbf{S}^{-1}$
4. Compute the correction: $\hat{\tilde{\mathbf{x}}} = \mathbf{K}\tilde{\mathbf{z}}$
5. Compute the state covariance matrix: $\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{KH})\mathbf{P}_{k+1|k}(\mathbf{I} - \mathbf{KH})^T + \mathbf{KVK}^T$

where $\mathbf{V}$ is the covariance matrix of visual measurement Gaussian noise. After obtaining the correction vector $\hat{\tilde{\mathbf{x}}}$, the nominal state of ESKF gets updated with the particular compositions with the correction state: $\mathbf{x} \leftarrow \mathbf{x} \oplus \hat{\tilde{\mathbf{x}}}$.

V.3 Delay Handler

In practical, the computational loading of VO algorithm is much heaver than ESKF core, therefore, it is a common situation that the time-stamp of odometry output is slower than current ESKF state. Hence, we introduce the sliding window method, proposed in [1, 3, 6], to handle the measurement delay issue. The concept of the method is to store the past states in a fixed size buffer. Once the delayed measurement is available, the update procedure will try to find the past state which has most similar time-stamp among buffer, then re-propagate all stored states after the one used for update.

VI Implementation

We take the ideas and concepts of [6, 23] as references to develop the software architecture of our VIO system. The whole package is developed by C++ under ARM based Linux environment. We adopt OpenCV library for image processing functions, especially the PnP solver with RANSAC iteration. For the matrix, vector and quaternion computing, the Eigen library is used for the reasons of numerical efficiency and its cross platform stability. We also employ the Boost library for multi-thread CPU computing for ESKF updates. The system provides the ROS wrapper for easily integration with other ROS packages. The result is fully open source with detail comments on core section in order to help other developer to understand quickly and be able to modify the codes depending on their demands. The package has been released on github link[4]. To get the best performance, the user has to run both camera intrinsic and camera-IMU extrinstic calibration as accurate as possible. The recommended package to execute this nonlinear optimization

The hardware overview is shown in Figure 1. We hacked a PS4-Eye[5], which is a low-cost, hardware synchronized, high resolution and FPS stereo camera, to be able to connect with Odroid XU4 through USB 3.0 port with modified firmware. The accelerometer and gyroscope data is obtained from mpu6050 (a type of low-cost MEMS IMU) which is installed on Pixhawk autopilot micro-controller. The total weight of overall system is lower than 250 grams, and the price of whole system is not higher than 150 dollars.

VII Experiments and Conclusion

VII.1 Experimental result

At present, the proposed VIO system has only been tested on hand-held scenario and EuRoC MAV Dataset. For the hand-held experiments, the performance highly depends on the quality of both intrinsic and extrinsic calibration, the complexity of texture and the motion behaviour. The ESKF propagation rate on Odroid XU4 could achieve 100Hz associated to Pixhawk IMU data publishing frequency.

For the MAV dataset, because the resolution of the image stream is too high for Odroid, the testing is executed on the desktop. The system performance is related to the parameter settings of VO algorithm.

VII.2 Conclusion and future direction

We propose a low-cost and lightweight loosely coupled visual inertial odomtery. The result is open source and easily to be implemented on any kinds of unmanned vehicle. Due to the benefits of forward facing stereo camera configuration, the platform which carries our approach could achieve obstacles avoidance without adding extra camera and be able to accomplish power-on-and-go scenario with no needs of particular initialization process.

Currently, the two most time consuming procedures among our VIO system are feature extraction and PnP problem solving. To improve the performance, the former task, from image processing view, could be divided into several cells. Then assigns each CPU thread to handle a subset of cells parallel. For the latter, if we could guarantee the correct rate of inputs feature matching pairs, the number of iteration and criteria of RANSAC procedure would be relaxed, thus, the VIO correction rate will increase.

[4] https://github.com/jim1993/StereoVIO

[5] https://www.playstation.com/en-us/explore/accessories/playstation-camera-ps4/

REFERENCES

[1] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[2] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 1885–1892, IEEE, 2016.

[3] N. de Palézieux, T. Nägeli, and O. Hilliges, "Duo-vio: Fast, light-weight, stereo inertial odometry," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 2237–2242, IEEE, 2016.

[4] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-imu-based localization: Observability analysis and consistency improvement," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 182–201, 2014.

[5] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 957–964, IEEE, 2012.

[6] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 3923–3929, IEEE, 2013.

[7] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pp. 225–234, IEEE, 2007.

[8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[9] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 15–22, IEEE, 2014.

[10] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*, pp. 834–849, Springer, 2014.

[11] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct slam with stereo cameras," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 1935–1942, IEEE, 2015.

[12] A. Concha, G. Loianno, V. Kumar, and J. Civera, "Visual-inertial direct slam," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 1331–1338, IEEE, 2016.

[13] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, p. 2005, 2005.

[14] S. M. Weiss, *Vision based navigation for micro helicopters*. PhD thesis, Citeseer, 2012.

[15] W. R. Hamilton, *Elements of quaternions*. Longmans, Green, & Company, 1866.

[16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564–2571, IEEE, 2011.

[17] M. Labbe and F. Michaud, "Appearance-based loop closure detection for online large-scale and long-term operation," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 734–745, 2013.

[18] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[19] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP Variants on Real-World Data Sets," *Autonomous Robots*, vol. 34, pp. 133–148, Feb. 2013.

[20] F. Fraundorfer, D. Scaramuzza, and M. Pollefeys, "A constricted bundle adjustment parameterization for relative scale estimation in visual odometry," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 1899–1904, IEEE, 2010.

[21] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[22] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, pp. 155–166, 2009.

[23] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*, Springer, July 2014.

Investigation on Natural Frequency and Fuselage Effect for Small UAVs Lateral Motion

M. El-Salamony,* S. Serokhvostov†
Moscow Institute of Physics and Technology
Department of Aeromechanics and Flight Engineering
140180, Gagarina street, 16, Zhukovsky, Russia

ABSTRACT

An accurate mathematical model is necessary for controlling an aircraft. Although the geometrical scale of Unmanned Aerial Vehicles (UAVs) is very small compared to the large aircrafts, they are usually designed by means of the procedure intended for large ones, and stability calculations similarly follow the same formulas. This fact can severely affect the basic assumptions of the formulas and hence it may not be suitable for UAVs. This research validates the dutch roll natural frequency of lateral motion calculated by comparing the usual methods of estimation for the manned aircraft found in references of Roskam and Ostoslavsky, and the numerical Vortex Lattice Method (VLM) program XFLR5 with experimental values of real flight. Also a study is carried out to examine the effect of fuselage on the dutch roll natural frequency to examine the possibility of neglecting it through the calculations. It is found that approximate methods for Roskam procedure is in accordance with the exact solution, and the same for Ostoslavsky. Estimation methods of Roskam (exact), Ostoslavsky and XFLR5 give good results in agreement with the experiment, while the approximate methods of Roskam underestimate the frequency. The contribution of the regular fuselage is found to be very small and it can safely be neglected.

NOMENCLATURE

ω_n	dutch roll mode natural frequency
θ	pitch angle
b	wing span
C_L	airplane lift coefficient in steady state condition
g	gravitational acceleration
I_{xx}	moment of inertia around fuselage axis
I_{yy}	moment of inertia around wing axis
I_{zz}	moment of inertia around normal to fuselage axis
L_β	roll angular acceleration per unit sideslip angle
L_p	roll angular acceleration per unit roll rate
L_r	roll angular acceleration per unit yaw rate
m	aircraft mass
N_β	yaw angular acceleration per unit sideslip angle
N_p	yaw angular acceleration per unit roll rate
N_r	yaw angular acceleration per unit yaw rate
N_{T_β}	yaw angular acceleration per unit sideslip angle (due to thrust)
S	wing reference area
U_1	cruise velocity
Y_β	lateral angular acceleration per unit sideslip angle
Y_p	lateral angular acceleration per unit roll rate
Y_r	lateral angular acceleration per unit yaw rate

*Email address: elsalamony.mostafa@phystech.edu
†Email address: serokhvostov@phystech.edu

1 INTRODUCTION

In order to design a control system for an aircraft, the main step is to make a mathematical model of flight mechanics for the aircraft. The controller's accuracy depends on the accuracy of the mathematical model with respect to the physical model. Even a simple PID control could be used to make the required response if the mathematical model was accurate enough.

According to flight mechanics, if any disturbance influences the aircraft (as gust wind or control surface deflection) the stable aircraft tries to damp this disturbance and return to its initial state. This behaviour is very important for the non-manoeuvrable aircraft and the knowledge about the stability is strongly required for the aircraft design and autopilot design.

For the conventional airplane shapes it is possible to separate

the disturbed motion on the longitudinal and lateral ones.
Lateral motion can be modelled as the fourth-order equation that describes three modes: spiral, roll and dutch roll. The first two modes are first-order (corresponding to the exponential decrease or increase) and the dutch roll is second-order (corresponding to the decreasing or increasing oscillations or exponential increasing/decreasing).
In case of damped vibrations, there are two definitions for the frequency: the damped frequency and the undamped/natural frequency. In this paper, the natural frequencies are considered.
Since the UAVs (including mini and micro) are usually designed according to the procedure used for the large aircrafts, the stability calculations also commonly follow the same formulas derived for the large aircrafts, but the geometrical scale of UAVs is much smaller. So, the forces acting on the UAV change their order of power nonlinearly and some assumptions for the manned aircrafts may be not valid in this case, and new assumptions can be introduced. One of the main questions is how the formulas for these frequencies change with the scale and Reynolds number.
First attempts of these investigations were conducted previously in [1]. Among all conclusions, the main finding was the possibility of separating the equations of UAVs disturbed motion into the longitudinal and lateral motions. In [2] a test case was studied for an UAV of mass of 150 gram, wing span of 85 cm. Based on the procedure of [3] it was found that the natural frequency of the short mode of longitudinal direction is big enough compared to the long mode. Longitudinal flight modes were investigated in detail in [4]. It is found that the natural frequency of the long mode can be predicted accurately by the exact methods used in [5] and [3] but the short mode was not captured due to the high damping ratio and the testing conditions. A method was recommended to overcome the high damping ratio in [6] by shifting the center of gravity (CG). Now the dutch roll oscillation of the lateral motion is being investigated in detail.
The goal of this investigation is to understand how accurate the calculations based on the "traditional" formulas with respect to the experimental values are. In this research, an investigation is carried out on the formulas and assumptions of calculating the natural frequency of the dutch roll mentioned by J. Roskam, D. Hull, and I. Ostoslavsky and compare their results with the numerical VLM calculations from XFLR5 which is mainly designed for small UAVs then these results are compared to real measurements of the natural frequencies obtained from UAV "Sonic 185" at flight.
For big aircrafts usually a mathematical model is created to simulate its response to follow predefined action in case of small air disturbance and the response is recorded in the form of flight path angles and compared to the oscillations from flight log of flight test to validate the mathematical model and in particular the frequency. Such a method is not applicable for small UAVs because the disturbances are relatively high compared to the forces applying on the UAV, that's why another method is proposed. Instead of comparing the data of flight angles obtained from the experiment and mathematical model, the flight angles are processed to obtain the main parameters of modelling (natural frequencies) and to compare them to the theoretical results. This method seems to be more appropriate for small UAVs flying in disturbed air.
Fuselage has essential role in the aircraft as it has to carry the weight and it is also affects the aerodynamic performance specially drag. Calculating the fuselage contribution analytically is not straight forward because of the shape complexity. Since drag force doesn't have big importance in calculating the natural frequencies of the aircraft, a study is conducted to quantify the effect of modelling of the fuselage to examine its importance for the scale of UAVs.

2 Investigated UAV

The UAV used in this research is "Sonic 185" of DYNAM [7]. Aircraft parameters and geometry are measured and listed in Tables 1 and 2.
Due to the absence of data about the airfoils used in the wing and empennage, another airfoil was estimated to have nearly similar profile shape from the known ones. Estimation of the airfoil is based on measurements of the thickness to chord ratio and the position of maximum camber and searching for a similar airfoil. Given that thickness to chord ratio is 10.7% at 39%, this leads to choose the airfoil E231 of Eppler series which shows good convergence as its thickness to chord ratio is 12.3% at 39.4%. For the tail unit, NACA 0006 is used.

Property	value
Mass $[kg]$	1.183
I_{xx} $[kg{\cdot}m^2]$	0.108
I_{yy} $[kg{\cdot}m^2]$	0.065
I_{zz} $[kg{\cdot}m^2]$	0.122
Cruise velocity $[m/s]$	8
Aspect ratio	10.295
Span $[m]$	1.85
Wing area $[m^2]$	0.33
Center of mass from leading edge of root section $[m]$	0.07

Table 1: Aircraft parameters.

3 Analytical Approach

Roskam and Ostoslavsky considered this task by two different procedures using dimensional and nondimensional stability parameters as shown below.

3.1 Roskam Procedure

Roskam's procedure [5] starts from estimating the aerodynamic coefficients (see Table 3), calculating the forces acting on the aircraft, then calculates the main characteristic

Property	Wing	Horizontal Tail	Vertical Tail
Aspect ratio	10.295	4.92	2.03
Root chord $[m]$	0.205	0.125	0.2
Tip chord $[m]$	0.06	0.02	0.115
Mean chord $[m]$	0.189	0.1	0.16
Span $[m]$	1.85	0.48	0.16
Area $[m^2]$	0.33	0.046	0.03
Sweep angle from leading edge $[degree]$	6.71	18.17	23.25

Table 2: "Sonic 185" geometry.

equation. Based on the big amount of experimental data, the coefficients are estimated taking into account many details which may increase the accumulative errors during calculations. The characteristic equation of the lateral motion [5]

Derivative	Sideslip Angle (β)	Roll Rate (p)	Yaw Rate (r)
Side Force Coeff. (C_y)	-0.17	0	0.14
Roll Moment Coeff. (C_l)	-0.113	-0.873	0.204
Yaw Moment Coeff. (C_n)	0.064	-0.127	0.124

Table 3: Aerodynamic lateral derivatives based on Roskam method.

is:

$$Ax^4 + Bx^3 + Cx^2 + Dx + E = 0 \tag{1}$$

where

$$A = U_1(1 - \bar{A}\bar{B}) \tag{2}$$

$$B = -Y_\beta(1 - \bar{A}\bar{B}) - U_1(L_p + N_r + \bar{A}N_p + \bar{B}L_r) \tag{3}$$

$$\begin{aligned} C = U_1(L_pN_r - L_rN_p) + Y_\beta(N_r + L_p + \bar{A}N_p \\ +\bar{B}L_r) - Y_p(L_\beta + N_\beta\bar{A} + N_{T\beta}\bar{A}) + U_1(L_\beta\bar{B} \\ +N_\beta + N_{T\beta}) - Y_r(L_\beta\bar{B} + N_\beta + N_{T\beta}) \end{aligned} \tag{4}$$

$$\begin{aligned} D = -Y_\beta(L_pN_r - L_rN_p) + Y_p(L_\beta N_r - N_\beta L_r \\ -N_{T\beta}L_r) - g\cos\theta_1(L_\beta + N_\beta\bar{A} + N_{T\beta}\bar{A}) \\ +U_1(L_\beta N_p - N_\beta L_p - N_{T\beta}L_p) - Y_r(L_\beta N_p \\ -N_\beta L_p - N_{T\beta}L_p) \end{aligned} \tag{5}$$

$$E = g\cos\theta_1(L_\beta N_r - N_\beta L_r - N_{T\beta}L_r) \tag{6}$$

where

$$\bar{A} = I_{xz}/I_{xx} \tag{7}$$

$$\bar{B} = I_{xz}/I_{zz} \tag{8}$$

After analysis, the results showed that natural frequency (ω_n) for dutch roll is 0.59 Hz. To simplify the decomposition of modes Roskam made an approximate solution for obtaining the natural frequency directly instead of solving the main fourth order equation by linking the forces directly to the natural frequency as follows:

$$\omega_{n1} = \sqrt{N_\beta + \frac{Y_\beta N_r - N_\beta Y_r}{U_1}} \tag{9}$$

Under the assumption that $(Y_\beta N_r - N_\beta Y_r)/U_1$ is significantly less than N_β, the former equation will be:

$$\omega_{n2} = \sqrt{N_\beta} \tag{10}$$

The natural frequency in these cases are ω_{n1}=0.5 Hz and ω_{n2}=0.49 Hz respectively.

3.2 Ostoslavsky Procedure

Ostoslavsky [3] has derived the characteristic equation by a different method. Instead of calculating the forces, the non-dimensional aerodynamic coefficients are used directly then the coefficients of the lateral characteristic equation are obtained. Using simple geometrical parameters, the aerodynamic coefficients can be estimated simply. Results are listed in Table 4.

Derivative	Sideslip Angle (β)	Roll Rate (p)	Yaw Rate (r)
Side Force Coeff. (C_y)	-1.494	0	0
Roll Moment Coeff. (C_l)	-0.011	-1.038	-0.089
Yaw Moment Coeff. (C_n)	0.036	0.073	0.032

Table 4: Aerodynamic lateral derivatives based on Ostoslavsky method.

Ostoslavsky introduced the characteristic equation of the system as fourth order function in its eigen values as:

$$F = \lambda^4 + a_1\lambda^3 + a_2\lambda^2 + a_3\lambda + a_4 = 0 \tag{11}$$

where the coefficients of these equations under the steady state condition are:

$$a_1 = -(0.5c_{L\beta} + \bar{c_{lp}} + \bar{c_{nr}}) \tag{12}$$

$$\begin{aligned} a_2 = c_{L\beta}/2(\bar{c_{lp}} + \bar{c_{nr}}) + (\bar{c_{lp}}\bar{c_{nr}} \\ -\bar{c_{lr}}\bar{c_{np}}) - \mu(\bar{c_{n\beta}} + \alpha\bar{c_{l\beta}}) \end{aligned} \tag{13}$$

$$a_3 = -\mu(\bar{c_{l\beta}}\bar{c_{np}} - \bar{c_{n\beta}}\bar{c_{lp}}) \tag{14}$$

$$\begin{aligned} a_4 = \mu c_L(\bar{c_{l\beta}}\bar{c_{nr}} - \bar{c_{n\beta}}\bar{c_{lr}} + \tan\theta_0) \\ (\bar{c_{l\beta}}\bar{c_{np}} - \bar{c_{n\beta}}\bar{c_{lp}})/2 \end{aligned} \tag{15}$$

where

$$\overline{c_{lp,lr,l\beta}} = \frac{c_{lp,lr,l\beta} m b^2}{4 I_{xx}} \tag{16}$$

$$\overline{c_{np,nr,n\beta}} = \frac{c_{np,nr,n\beta} m b^2}{4 I_{yy}} \tag{17}$$

$$\mu = \frac{2m}{\rho S c} \tag{18}$$

For simplification of the analysis, the following approximate equation can be used:

$$\omega_{n1} = -\frac{a_3}{\bar{c_{lp}} 2\pi\tau} \tag{19}$$

where

$$\tau = \frac{2m}{\rho S U_1} \tag{20}$$

The results of the exact method show highly-damped situation while the approximate method showed that ω_{n1}=0.556 Hz.

4 Numerical Approach

In numerical methods as Vortex Lattice Method (VLM) – as used in XFLR5 – the aircraft is divided into small panels. For each panel a combination of source, sink, and vortex is added in one quarter of the panel, and a control point is added after three quarters of the panel to achieve the no-penetration condition [8]. By solving *N* equations obtained from the *N* panels, the vortex strength is determined for each panel then the normal and tangential forces acting on the aircraft are obtained then converting them into non-dimensional coefficients. The next step is to import these values – which depend on the angle of attack and velocity – into the state space matrix and obtain the eigen values of the matrix which are a combination of natural frequency and damping ratio and they can be separated easily. XFLR5 is used because it is open source and used widely for UAV design process and also has the ability to obtain the natural frequencies and damping ratios directly.
The UAV is plotted using the measurements from Tables 1 and 2 as illustrated in Figure 1. It is noted that fuselage in this case has is not modelled.

Calculation done showed that ω_n=0.57 Hz.

5 Experimental Approach

The experiment was conducted for the steady cruise flight with some excitation for the dutch roll mode. To prevent additional disturbances, the plane was controlled in manual mode without autopilot stabilization. During the flight the aircraft was balanced so as it moves straight with constant altitude and constant velocity. Flight parameters were controlled form the ground station of Ardupilot by means of telemetry link. Flight data are obtained from the Inertial Measuring

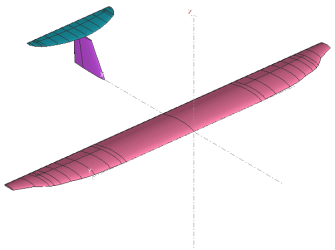


Figure 1: Sonic 185 drawing in XFLR5.

Unit (IMU) of the "ArduPilot Mega" autopilot which measure pitch, roll, and yaw angles of the aircraft and has sampling frequency of 3.7 Hz.
The retrieved data are processed by Fast Fourier Transform [9] once without filter and another time with filter using MATLAB. While examining the signal without filter, it is taken into account the signal first and last points have the same values to prevent aliasing.
Data are filtered by Hanning filter [10, 11] to prevent leakage in the transform [12]. Such filter is chosen for this case because:

- overcome the noise and get the mean value of the frequency,
- the exact amplitude of the frequency is not as important as the value of the frequency itself,
- the investigated signal is random and have unknown frequency,
- the vibrations are within narrow band.

For these four reasons, the most suitable filter is Hanning filter [12]. Sample of measured yaw angle is shown in Figure 2. Fast Fourier Transform is used in converting discrete time samples from time to frequency domain. After processing and filtering, the results show a freq of 0.61±0.06 Hz as shown in Figures 3–5 that show samples of the obtained results at different periods of time.

6 Fuselage Effect

Fuselage is one of the main parts of the aircraft which have direct influence on the behaviour of the aircraft towards disturbances.
Effect of the fuselage contributes in two effects: inertial forces and aerodynamic forces. Inertial forces are critical and cannot be omitted because the fuselage is essential source of weight and its inertia is essential. Aerodynamic forces applied on the fuselage has an influence on the total

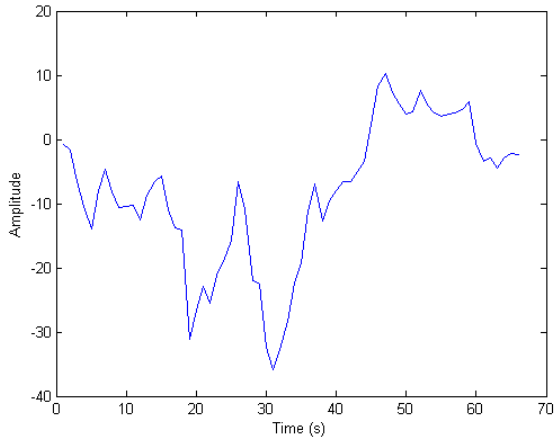


Figure 2: Sample of yaw angle recorded during flight.

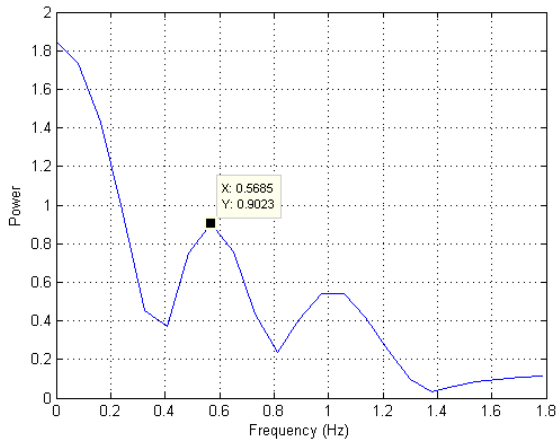


Figure 3: Sample 1 of dutch roll mode frequency.

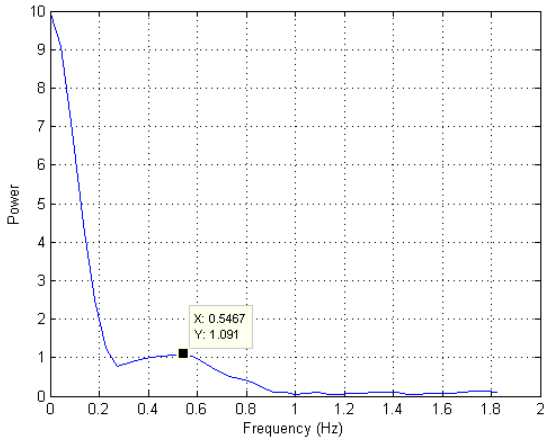


Figure 4: Sample 2 of dutch roll mode frequency.

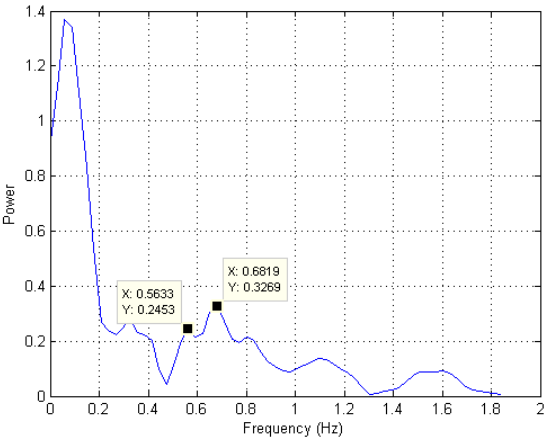


Figure 5: Sample 3 of dutch roll mode frequency.

forces and moments applied on the aircraft which determine its behaviour towards any disturbances. Calculation of aerodynamic and stability parameters for fuselage in the theoretical methods is complex and needs many information about the exact layout which is not available yet in the preliminary design phase. On the other hand, many VLM codes need much less information and an approximate layout is enough to estimate the applied forces. This conflict is the motivation to investigate the influence of the fuselage aerodynamic effect in calculation of natural frequency for the dutch roll mode. This mode is selected especially because it is quite popular that modeling of the fuselage is important is mandatory / important for lateral motion, and now this lemma is being criticized.

To investigate the fuselage aerodynamic influence on the dutch roll natural frequency, the frequency is examined by comparing two UAV cases: without and with fuselage using XFLR5. The second configuration, shown in Figure 6, is calculated in the same way explained in Section 4.

Results show that ω_{n1}= 0.582 Hz for the case with fuselage.

7 Analysis and Discussion

It is noticed that the aerodynamic coefficients estimated from the procedures of Roskam and Ostoslavsky are not matched together, for example the parameter $c_{y\beta}$ in Ostoslavsky method is ten times higher than Roskam. Though, there is good agreement in the natural frequencies. From here it is concluded that each method must use its own aerodynamic and stability coefficients.

The exact method of Ostoslavsky shows high damping ratio while the approximate solutions give results in accordance with the experimental one, which means that the assumptions used are still valid for UAVs. On the other hand, approximate

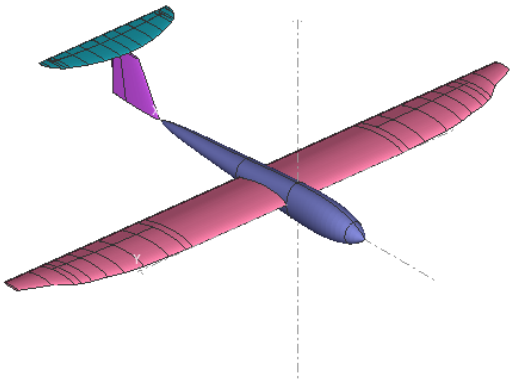


Figure 6: Sonic 185 including fuselage in XFLR5.

methods of Roskam don't achieve the required accuracy, and hence the assumptions he did are invalid for the case of small UAVs.
Calculations and experiments of this research – beside [4, 6]) – show that the frequencies for the longitudinal and lateral modes are different, so it is possible to separate the longitudinal and lateral disturbed motions for the aircrafts of the type investigated. Also the dutch roll mode can be sorted out from the whole lateral motion.
By comparing the results of the exact method of Roskam, approximate method of Ostoslavsky and XFLR with the experiment, it is notable that these methods give result in the same order of the actual one as shown in Table 5 while the approximate methods of Roskam are not accurate enough. This means that a set of methods valid for the large aircrafts can be implemented to the smaller ones (for lower Re numbers and small geometric scales).

Method	Frequency (Hz)
Roskam – exact	0.59
Roskam – approx (1)	0.50
Roskam – approx (2)	0.49
Ostoslavsky – exact	–
Ostoslavsky – approx	0.56
XFLR5 without fuselage	0.57
XFLR5 with fuselage	0.58
experiment	0.61 ± 0.06

Table 5: Results of dutch roll natural frequency using the different methods.

Formulas and experimental data show that the value of damping ratio is very close to the one corresponding to aperiodical motion. In this case to define the mode realizing (periodic or aperiodic) rather precise values of necessary parameters are required. As we can't grantee the absolute precision of parameters' values used in the calculations this can explain the fact that exact Otoslavsky formula predict overdamped (aperiodic motion) while approximate Ostoslavsky formula gives periodic damped motion.
Considering the fuselage effect on the dutch roll frequency, the difference between the cases with and without fuselage is 0.028 Hz, which is can be neglected compared to the uncertainty of the experiment (0.06 Hz). This means that absence of conventional fuselage will not affect the accuracy severely and it can be neglected while studying the natural frequency of the lateral mode.

8 Conclusion

This research validates the dutch roll natural frequency of lateral motion calculated by the usual methods of estimation for the manned aircraft found in the references of Roskam and Ostoslavsky, and the numerical VLM program XFLR5 with experimental values of real flight. It is found that exact and approximate methods of Ostoslavsky, exact method of Roskam, and XFLR5 estimate the frequency within range of the experimental results while the approximate methods of Roskam underestimates the frequency and hence the assumptions used are not valid in case of small UAVs. Results from analytical methods are valid only for the aerodynamic coefficients defined in the same procedure.
The methods for the large aircraft dynamics of the disturbed motion can be implemented to the smaller aircrafts and lower Re numbers while the assumptions must considered carefully taking into consideration that some stability coefficients cannot be neglected as in the case of large aircraft.
The role of fuselage in natural frequency estimation is examined and it is found that absence of conventional fuselage will not affect the accuracy severely and it can be neglected.

References

[1] S.V. Serokhvostov, A.A. Dmentyev, and T.E. Churkina. Analytical investigation of mav dynamical characteristics and its influence on design. In *International Micro Air Vehicle Conference (IMAV), Braunschweig, Germany*, 2010.

[2] S. Serokhvostov, N. Pushchin, and K. Shilov. Mav unsteady characteristics in-flight measurement with the help of smartap autopilot. In *International Micro Air Vehicle Conference (IMAV), Toulouse, France*, 2010.

[3] IV Ostoslavsky. *Aerodynamics of aircraft*. Oborongiz, Moscow, 1957.

[4] M. El-Salamony, S. Serokhvostov, A. Epikhin, and K. Zaripov. Comparison between experimental, analytical, and numerical methods in natural frequencies of the longitudinal flight modes for small uavs. In *International Micro Air Vehicle Conference (IMAV), Beijing, China*, 2016.

[5] Jan Roskam. *Airplane flight dynamics and automatic flight controls*. DARcorporation, 1995.

[6] M. El-Salamony, S. Serokhvostov, A. Epikhin, and K. Zaripov. Measurement and calculation of natural frequencies of the longitudinal flight modes for small uavs. In *Extremal and Record-Breaking Aircrafts Workshop (ERBA), Zhukovsky, Russia*, 2016.

[7] Universal Power Enterprizes Ltd. Sonic 185: Instruction manual. *Universal Power Enterprizes Ltd., Kowloon, Hong Kong*, 2008.

[8] Edward Lewis Houghton and Peter William Carpenter. *Aerodynamics for engineering students*. Butterworth-Heinemann, 2003.

[9] Richard S Figliola and Donald Beasley. *Theory and design for mechanical measurements*. John Wiley & Sons, 2015.

[10] L Dactron. Understanding fft windows. *Application Note. lds group*, 2003.

[11] Tim Wescott. Sampling: what nyquist didnt say, and what to do about it. *Wescott Design Services, Oregon City, OR*, 2010.

[12] Pierre Wickramarachi. Effects of windowing on the spectral content of a signal. *Sound and vibration*, 37(1):10–13, 2003.

Team MAVion entry in the IMAV'17 outdoor challenge – A tail-sitting trajectory-tracking μUAV

Leandro R. Lustosa[1], Jacson M. O. Barth[2], Jean-Philippe Condomines[2], François Defaÿ[1], Jean-Marc Moschetta[1] *
[1]Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO), Toulouse 31400, France
[2]École Nationale de l'Aviation Civile (ENAC), BP 54005, Toulouse Cedex 4, 31055, France

ABSTRACT

This paper outlines current research conducted on tilt-body micro air vehicles at ISAE, and how we exploit recent advances to provide a tail-sitting flying-wing entry for the IMAV'17 outdoor challenge capable of performing automatic vertical take-off, landing, and trajectory-tracking.

1 INTRODUCTION

1.1 A quick contextualization

Since their debut in the 50s, tail-sitting vehicles would only be flown by the most experienced pilots. Recent advances on microelectromechanical (MEMs) inertial sensors and embedded computing, on the other hand, support stability augmentation systems (SAS) in mitigating unstable dynamic modes and allowing for inexperienced (or even autonomous) flight. A large and growing body of literature [1, 2] has investigated underlying modeling, control and planning issues specific to this architecture. Modeling, for instance, has recently called the attention of aerodynamicists due to unfamiliar high incidence propeller operation [3]. Further striking features are its nonlinearity, underactuation and difficult-to-model post-stall aerodynamics, which pose a beautiful challenge for the practicing control and robotics communities [4].

From a practical perspective, far too little attention has been paid to a fundamental question: how should the pilot control such a multifaceted architecture? While it seems appropriate to fly it as a quadrotor when in hover and as a fixed-wing when in cruise, it is unclear how to command it during transition. While platforms that provide an automatic transition phase do not face such dilemma, our control architecture allows for steady-flight in each forward velocity from rest to full forward speed and such question is pertinent. We explore this issue in this paper, and provide an entry to the IMAV'17 competition by applying an adequate control law to our piloting philosophy. We additionally show that guidance law design is independent of vehicle dynamics in our proposed architecture.

1.2 IMAV outdoor challenge specs

The MAVion will take up the outdoor flight performance challenge. It comprises taking-off automatically, flying as many laps as possible around 2 poles during a single flight, and landing automatically (see Fig. 1).



Figure 1: Competition lay-out at the Francazal airport runway.

2 THE MAVION PROJECT IN A NUTSHELL

2.1 A brief history

ISAE started its tilt-body research with a tail-sitter called *Vertigo* that was developed and flown in 2006. Miniaturization of the Vertigo was conducted at University of Arizona to provide the *Mini-Vertigo*, a 30cm span coaxial-rotor MAV capable of transition flight. However, the coaxial rotor driving mechanism represented an additional weight and precluded further miniaturization. Moreover, the Mini-Vertigo suffered from fairly high induced drag in cruise flight due to its low aspect ratio. In view of improving aerodynamic performance in forward flight and simplifying the rotor mechanism, a new tilt-body configuration based on bimotor flying-wings was proposed. The MAVion was initially designed to be a reasonably efficient airplane capable of flying outdoors and requiring a minimum number of moving parts (e.g., no tilting wings, no tilting motors). The main design guidelines were simplicity and transition flight robustness. Currently, our prototypes fly by means of either nonlinear inversion or scheduled-LQR control laws. In parallel, a *Roll and Fly* [5]

*Corresponding e-mail address: l.lustosa@isae.fr

version with wheels was designed to pursue indoors exploration.

2.2 Airframe

Fig. 2 illustrates the proposed airframe. In short, the MAVion is a flying-wing capable of tail-sitting. Moreover, it is capable of sustaining trimmed flight from rest up to 25m/s [6]. Equivalently, it sustains hover flight in face of external wind up to 25m/s (see Fig. 3). The MAVion has a 42cm wingspan and 2:1 aspect ratio. It weights (airframe, avionics and batteries) 435g and it is expected to endure a 10-30min flight – depending on mission hovering/cruise ratio requirements – based on wind tunnel measurements [7].

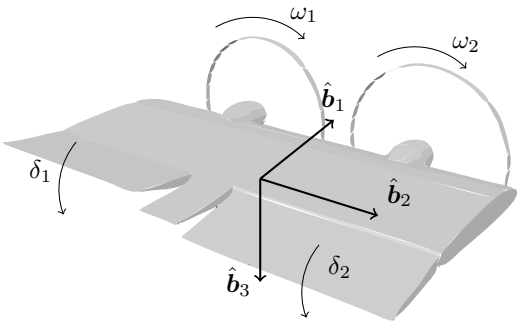


Figure 2: Perspective view, body-axis definition and actuation inputs for the MAVion tail-sitting vehicle. (Adapted from [7].)

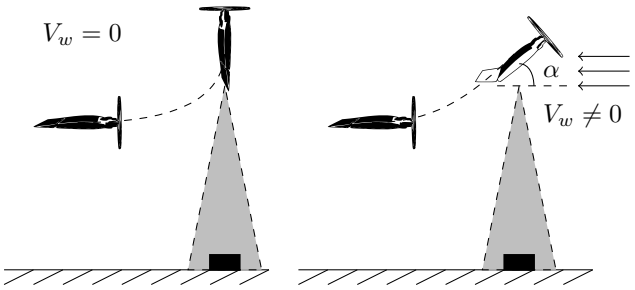


Figure 3: Transition maneuvers and hovering over ground target under non-windy (left) and windy (right) conditions. (Adapted from [8].)

2.3 Avionics

The competition version of the MAVion airframe is equipped with the *Paparazzi Apogee* autopilot board. It contains low-cost 3-axis accelerometers, rate-gyros and magnetometer for navigation purposes, and a MicroSD card slot for logging flight data. The autopilot communicates with an external GNSS receiver, a pitot tube and two radio links (data-link and fail-safe RC control). Digital servos govern elevons while DC motors control propellers motion.

Although not yet natively compatible with convertible architectures at the time of writing this paper, the Paparazzi framework [9] is modular, provides various handy mathematical C libraries, and encourages third-party modifications, such as the guidance, navigation and control techniques proposed herein.

2.4 Mathematical model

We assume MAVion dynamics as described by the state space nonlinear equations set forth in [7] and represented hereafter by $\gamma(\cdot)$ as follows

$$\dot{\boldsymbol{x}} = \gamma(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^{10}$, $\boldsymbol{u} \in \mathbb{R}^4$, $\boldsymbol{w} \in \mathbb{R}^3$ are, respectively, MAVion state, control inputs and wind disturbances, given by

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{v}_l & \boldsymbol{\omega}_b & q \end{pmatrix}^T \tag{2}$$

and

$$\boldsymbol{u} = \begin{pmatrix} \omega_1 & \omega_2 & \delta_1 & \delta_2 \end{pmatrix}^T \tag{3}$$

where $\boldsymbol{v}_l$, $\boldsymbol{\omega}_b \in \mathbb{R}^3$, $q \in \mathbb{R}^4$, denote, respectively, linear velocity described in local NED frame, angular velocity in body frame, and vehicle attitude in quaternion formulation. The control input $\boldsymbol{u}$ components and its associated sign conventions are depicted in Fig. 2.

The structure of $\gamma(\cdot)$ is fairly complex and, consequently, we point the interested reader to [7] for detailed equations and assumptions. Nevertheless, we remark that $\gamma(\cdot)$ provides an analytic continuous singularity-free model over a full 360° angle-of-attack and sideslip flight envelope. Additionally, the model incorporates fundamental nonlinear aerodynamics – e.g., post-stall and prop-wash effects – in view of hybrid and high maneuverable vehicles. Incidentally, the tilt-body nature of the vehicle calls for a global numerically stable formulation of attitude and upholds quaternion employment. Another important practical feature of $\gamma(\cdot)$ is its polynomial-like algebraic structure that allows for efficient online trajectory generation by means of semidefinite programming and sum-of-squares (SOS) optimization. The IMAV outdoor challenge, however, proposes a known *a priori* trajectory and, therefore, we benefit from a simpler guidance solution detailed in Section 5.

In summary, for this challenge, whereas $\gamma(\cdot)$ is not used for trajectory generation, it supports simulation and control design (see Section 4). Our MATLAB-based simulator is composed of an outdoor mode which simply integrates (1) numerically, and an indoor mode that relies on the Unreal Engine (UE) physics collision engine. Fig. 5 illustrates the overall architecture: MATLAB computes resultant forces and moments based on γ and sends it to UE for kinematics computation in view of a UE-based map (with obstacles). If existent, collisions are handled and an updated state is then sent back to MATLAB for computing new forces. Additionally, UE provides a beautiful 3D viewer (see Fig. 4) and straightforward map/obstacles editing tools.

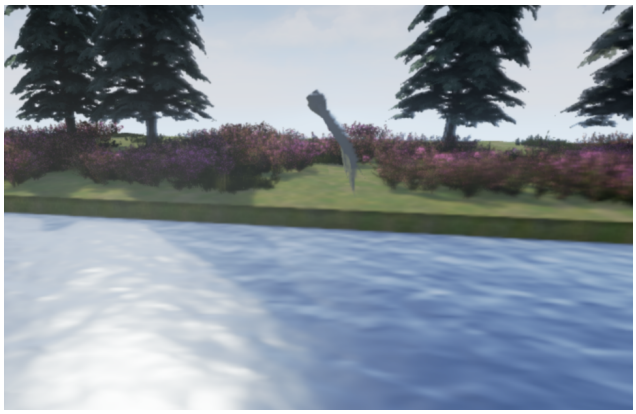


Figure 4: MAVion Unreal Engine-based simulator.

3 NAVIGATION SYSTEM

In light of the challenge requirements and available on-board avionics, we opted for deploying the subset of navigation sensors described in Table 1. We employ dead-reckoning inertial navigation assuming stationary and flat Earth on account of low-grade inertial sensors. A magnetometer/accelerometer-based complementary filter (CF) bounds the resulting divergent attitude errors. On top of the aforementioned system, a loosely-coupled extended Kalman filter (EKF) corrects for position and velocity errors by means of GNSS receiver measurements. Such architecture (see Fig. 6) benefits from a previous [10] compelling filter formulation that accounts for complementary filter dynamics in the EKF state transition matrix. This section briefly details the overall architecture for the sake of completeness. For a detailed study, we invite the reader to read [10].

Function	Device	Noise (typ)	Bias (typ)
Rate-gyro	MPU-9150	0.005 ($^o/s/\sqrt{Hz}$)	20 ($^o/s$)
Accelerom.	MPU-9150	400 ($\mu g/\sqrt{Hz}$)	150 (mg)
Magnetom.	MPU-9150	N/A	N/A
GNSS pos.	NEO-6M	$\sigma = 2.5$ (m)	0 (m)
GNSS vel.	NEO-6M	$\sigma = 0.1$ (m/s)	0 (m/s)

Table 1: Avionics subset for loosely-coupled GNSS and magnetometer aided strapdown inertial navigation. Inertial sensors specifications are readily determined from manufacturer datasheet. On the other hand, our GNSS receiver manufacturer datasheet lacks noise and bias statistics (understandably, since GNSS receiver loosely-coupled position and velocity errors are hardly Gaussian). Therefore, the available information was loosely recast to somewhat equivalent Gaussian noise and bias quantities (in view of suboptimal Kalman filtering implementation).

We model and simulate rate-gyro measurements $\boldsymbol{\omega} \in \mathbb{R}^3$

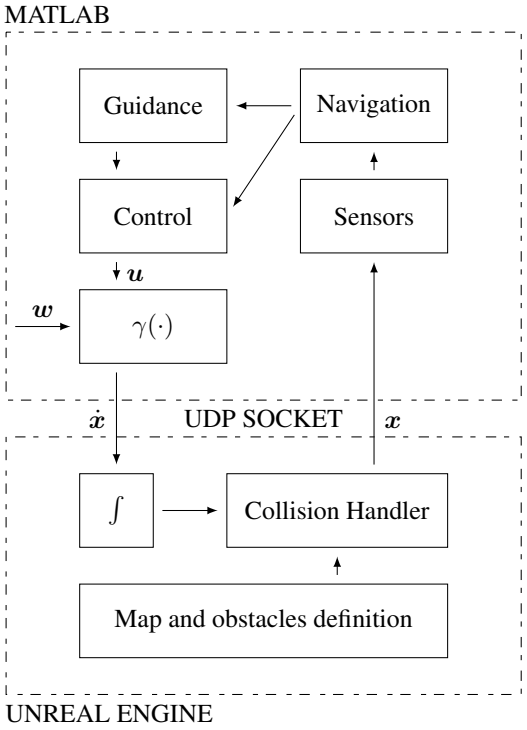


Figure 5: MAVion simulator architecture.

according to

$$\boldsymbol{\omega}(t) = \boldsymbol{\omega}_b^b(t) + \boldsymbol{\varepsilon}_b + \boldsymbol{\nu}_b^g(t) \tag{4}$$

where $\boldsymbol{\omega}_b^b$, $\boldsymbol{\varepsilon}_b \in \mathbb{R}^3$, and $\boldsymbol{\nu}_b^g \sim N(\mathbf{0}, \Sigma_g)$ denote, respectively, nominal angular velocity, rate-gyro drift and rate-gyro white Gaussian noise (all described in body frame). Likewise, accelerometer and magnetometer measurements, namely $\boldsymbol{f}(t)$ and $\boldsymbol{b}(t)$ are modeled as

$$\boldsymbol{f}(t) = \boldsymbol{a}_b^b(t) - \boldsymbol{g}_b(t) + \boldsymbol{\nabla}_b + \boldsymbol{\nu}_b^a(t) \tag{5}$$

and

$$\boldsymbol{b}(t) = \boldsymbol{B}_b(t) + \boldsymbol{\Delta}_b + \boldsymbol{\nu}_b^m(t) \tag{6}$$

where $\boldsymbol{a}_b^b$, $\boldsymbol{g}_b$, $\boldsymbol{\nabla}_b \in \mathbb{R}^3$, and $\boldsymbol{\nu}_b^a \sim N(\mathbf{0}, \Sigma_a)$ denote, respectively, linear acceleration, local gravity field, accelerometer bias and noise. Additionally, $\boldsymbol{B}_b$, $\boldsymbol{\Delta}_b \in \mathbb{R}^3$, and $\boldsymbol{\nu}_b^m \sim N(\mathbf{0}, \Sigma_m)$, denote local magnetic field, magnetometer bias and noise.

Rate-gyro measurements are filtered to bound attitude errors by means of the complementary filter, in such a way that the computed dead-reckoning angular velocity is given by

$$\boldsymbol{\omega}_c^c = \boldsymbol{\omega} - k_a \boldsymbol{f} \times D_c^l \boldsymbol{g}_l + k_m \boldsymbol{b} \times D_c^l \boldsymbol{B}_l \tag{7}$$

where k_a, $k_m \in \mathbb{R}_+$, and $D_c^l \in SO(3)$, are the complementary filter gains and NED-to-body CF-computed direction cosine matrix. Subsequently, $\boldsymbol{\omega}_c^c$ and $D_l^c \boldsymbol{f} + \boldsymbol{g}_l$ are integrated (in

the rigid-body motion sense) to yield CF-computed estimates of quaternion attitude, NED-position and velocity, namely, q_c^l, $\boldsymbol{p}_l^c$ and $\boldsymbol{v}_l^c$ (see Fig. 6).

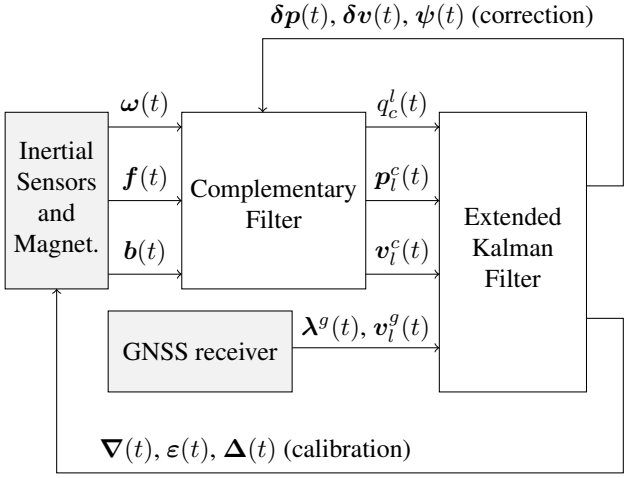


Figure 6: CF-EKF filter architecture. (Adapted from [10].)

Notice that CF outputs estimates with errors due to sensor imperfections. Previous work [10] modeled how CF errors evolve with time in function of sensor imperfections. That is, given unknown CF errors and sensor biases, namely,

$$\boldsymbol{x}_{EKF} = \begin{pmatrix} \boldsymbol{\delta p}_l & \boldsymbol{\delta v}_l & \boldsymbol{\psi} & \boldsymbol{\nabla}_b & \boldsymbol{\varepsilon}_b & \boldsymbol{\Delta}_b \end{pmatrix}^T \tag{8}$$

and process noise as

$$\boldsymbol{w}_{EKF} = \begin{pmatrix} \boldsymbol{\nu}_b^a & \boldsymbol{\nu}_b^g & \boldsymbol{\nu}_b^m & \boldsymbol{a}_b^b \end{pmatrix}^T \tag{9}$$

[10] computes $g(\cdot)$ such that

$$\dot{\boldsymbol{x}}_{EKF} = g(\boldsymbol{x}_{EKF}, \boldsymbol{w}_{EKF}) \tag{10}$$

The jacobians of $g(\cdot)$ yield the state transition matrix F_k and process noise transport matrix B_k that are employed in an EKF for CF error estimation. Navigation errors $\boldsymbol{\delta p}_l$, $\boldsymbol{\delta v}_l$, $\boldsymbol{\psi}$ are periodically used for correcting the CF while $\boldsymbol{\nabla}_b$, and $\boldsymbol{\varepsilon}_b$ and $\boldsymbol{\Delta}_b$ are periodically used for sensors in-flight calibration (see Fig. 6).

4 Control system

Our approach is to regulate the MAVion to *a priori* computed equilibrium points and limit cycles of $\gamma(\cdot)$ by means of stabilization of unstable transverse dynamics [11] employing scheduled-LQR control. Although often overlooked, we remark that any quaternion-based model linearization yields uncontrollable Jacobians [12] that preclude LQR direct employment. We overcome this shortcoming by employing the strategies set forth in [12] (a commonplace strategy, for instance, is to neglect one out of the four quaternion components in LQR design). Furthermore, quaternion double cover of $SO(3)$ does not pose a problem since we are applying local controllers and reference-to-estimated quaternions unwinding is effortlessly detected by dot product sign inspection.

4.1 Longitudinal equilibrium points

For longitudinal flight, previous work [6] experimentally shows that trimmed flight condition establishes an one-to-one correspondence between pitch angles θ and free-stream velocities $\boldsymbol{v}$. This motivated us to pursue a velocity controlled architecture, where the pilot (or high level guidance loop) does not command angles or angle rates but, instead, desired velocities in an almost-body frame (i.e., forward horizontal, lateral horizontal and vertical velocities). Therefore, a given desired velocity $\boldsymbol{v}_0$ is sufficient to unambiguously define the remaining of the desired equilibrium point $\boldsymbol{x}_0$ (e.g., attitude) in longitudinal flight by means of the (unique) solution of

$$\gamma\left((\boldsymbol{v}_0, \boldsymbol{0}, \boldsymbol{q}), \boldsymbol{u}, \boldsymbol{0}\right) = \boldsymbol{0} \tag{11}$$

The Jacobian of γ evaluated at $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$ yields an uncontrollable linear system (A, B). After it is rendered controllable by means of the techniques in [12], we design a LQR controller $\Delta\boldsymbol{u} = -K\Delta\boldsymbol{x}$ to minimize

$$J(Q, R) = \int_0^\infty \left(\Delta\boldsymbol{x}^T Q \Delta\boldsymbol{x} + \Delta\boldsymbol{u}^T R \Delta\boldsymbol{u}\right) dt \tag{12}$$

given an appropriate choice of Q and R. These are tuned by trial-and-error runs in our simulator to account for actuator bandwidth, state estimation imperfections and embedded computer sampling times (all comprised in simulation). Once appropriate Q and R are found for hover operation, they are replicated for all other equilibrium points.

4.2 Lateral limit cycles

In contrast to longitudinal flight, lateral dynamics lacks equilibrium points in view of our choice for $\boldsymbol{x}$. One way to see this is observing that any constant $\boldsymbol{\omega}_b \neq \boldsymbol{0}$ changes q periodically. Therefore, instead of searching for equilibrium points, we search for states $\boldsymbol{x}$ such that the lateral force constitutes a coordinated curve centripetal force while maintaining constant altitude, that is,

$$\gamma\left((\boldsymbol{v}_b, \boldsymbol{\omega}_b, q), \boldsymbol{u}, \boldsymbol{0}\right) = \begin{pmatrix} \boldsymbol{\omega}_b \times \boldsymbol{v}_b \\ \boldsymbol{0} \\ \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}_b^T \\ \boldsymbol{\omega}_b & -[\boldsymbol{\omega}_b\times] \end{bmatrix} q \end{pmatrix} \tag{13}$$

In this way, our search for limit cycles reduces to a nonlinear root solving problem. Once limit cycles are found for a myriad of $(\boldsymbol{v}, \boldsymbol{\omega})$, LQR control can be designed by means of previous (Q, R) and transverse dynamics technology in [11].

Notice that we assume a bijection between equilibrium $(\boldsymbol{v}, \boldsymbol{\omega})$ and the other remaining variables[1]. This is a result which has not yet been established, but it is intuitively reasonable. Further study is required to establish its validity.

[1] Bijection in the sense of classes of equivalence: that is, the set of states $\boldsymbol{x}$ belonging to a same limit cycle composes a class of equivalence, and this class is in an one-to-one correspondence to $(\boldsymbol{v}, \boldsymbol{\omega})$ reference commands.

4.3 RC piloting interface

Notice that control references (equilibrium points or limit cycles) are given in terms of desired velocities $(\boldsymbol{v}, \boldsymbol{\omega})$. This means that a human pilot controls the MAVion thinking on where it needs to go instead of which orientation it needs to be. This abstracts the internal workings of the MAVion and relieves piloting efforts. Fig. 7 illustrates our proposition of RC controller input assignment for commonplace RC radio standards.

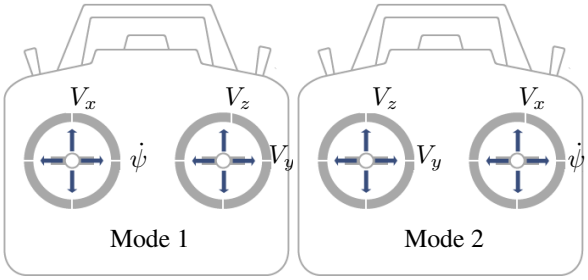


Figure 7: Input assignments of standard RC radio modes. Notice that ψ, V_x, V_z and V_y denote, respectively, desired heading with respect to geographic North, forward velocity, vertical velocity and lateral velocity.

5 Guidance system

We employ the commonplace guidance strategy described in [13], which is fully compatible with our aforementioned velocity-based controller architecture. This strategy is readily implemented in most open-source fixed-wing autopilots (e.g., `Pixhawk`, `Paparazzi`) and can be easily adapted for our purposes in view of our choice of control system reference parametrization – namely, forward and turn velocities (v, ω).

6 Simulation results

Flight simulations were conducted employing the aforementioned guidance, navigation and control strategies. The MAVion model is assumed known but we add wind disturbances up to 5m/s to account for robustness in control design. Additionally, navigation system routines are computed assuming corrupted sensor measurements according to Table 1. Figure 8 illustrates the obtained results. Those were obtained after a few controller tuning iterations. While it took considerable effort to find adequate LQR tuning weights, a promising result – in terms of performance and robustness – was found.

7 Conclusion

Previous research conducted on tilt-body vehicles at ISAE lays the groundwork for our IMAV competition entry. This paper provides general information on our architecture and shows promising results in simulation.

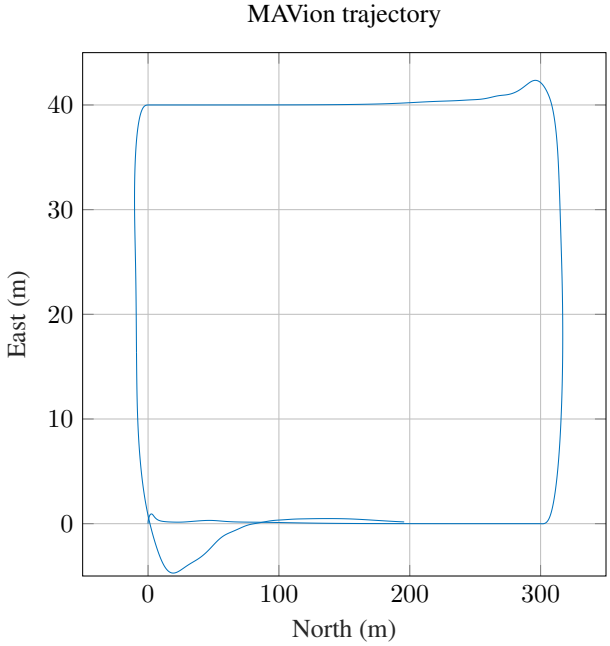


Figure 8: Simulation results: MAV trajectory.

Acknowledgements

The authors gratefully acknowledge the Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, (Brazilian National Science Foundation), for partial financial support for this work through the "Ciência sem Fronteiras" program.

References

[1] S. Verling, B. Weibel, M. Boosfeld, K. Alexis, M. Burri, and R. Siegwart. Full attitude control of a VTOL tailsitter UAV. In *IEEE International Conference on Robotics and Automation*, 2016.

[2] T. Matsumoto, K. Kita, R. Suzuki, A. Oosedo, K. Go, Y. Hoshino, A. Konno, and M. Uchiyama. A hovering control strategy for a tail-sitter VTOL UAV that increases stability against large disturbance. In *IEEE International Conference on Robotics and Automation*, 2010.

[3] M. Bronz and A. Drouin. Preliminary design estimation of the V/STOL airplane performance. In *International Micro Air Vehicles Conference and Flight Competition*, 2015.

[4] Joseph Moore, Rick Cory, and Russ Tedrake. Robust post-stall perching with a simple fixed-wing glider using LQR-trees. *Bioinspiration and Biomimetics*, 9(2), 2014.

[5] P. A. Alheritiere, R. Olivanti, L. R. Lustosa, F. Defaÿ, and J.-M. Moschetta. Nonlinear control of a particular

tilt-body MAV: The Roll and Fly. In *The 24th Mediterranean Conference on Control and Automation*, 2016.

[6] M. Itasse, J.-M. Moschetta, Y. Ameho, and R. Carr. Equilibrium transition study for a hybrid MAV. *International Journal of Micro Air Vehicles*, 3(4):229–246, 2011.

[7] L. R. Lustosa, F. Defaÿ, and J.-M. Moschetta. Longitudinal study of a tilt-body vehicle: modeling, control and stability analysis. In *International Conference on Unmanned Aircraft Systems*, 2015.

[8] L. R. Lustosa, F. Defaÿ, and J.-M. Moschetta. Development of the flight model of a tilt-wing micro-air vehicle. In *International Micro Air Vehicle Conference and Competition*, 2014.

[9] G. Hattenberger, M. Bronz, and M. Gorraz. Using the paparazzi UAV system for scientific research. In *International Micro Air Vehicle Conference and Competition*, 2014.

[10] L. R. Lustosa, S. Pizziol, F. Defaÿ, and J.-M. Moschetta. An error model of a complementary filter for use in Bayesian estimation - The CF-EKF filter. In *20th IFAC Symposium on Automatic Control in Aerospace*, 2016.

[11] I. R. Manchester. Transverse dynamics and regions of stability for nonlinear hybrid limit cycles. In *18th IFAC World Congress*, 2011.

[12] L. R. Lustosa, F. Cardoso-Ribeiro, F. Defaÿ, and J.-M. Moschetta. A new look at the uncontrollable linearized quaternion dynamics with implications to LQR design. 2018 (to be published).

[13] Sanghyuk Park, John Deyst, and Jonathan How. A new nonlinear guidance logic for trajectory tracking. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Guidance, Navigation, and Control*, 2004.

Simulation and Control of a Tandem Tiltwing RPAS Without Experimental Data

Y. Beyer*, T. Krüger, A. Krüger, M. Steen, P. Hecker
Institute of Flight Guidance, TU Braunschweig, Hermann-Blenk-Str. 27, 38108 Brunswick, Germany

ABSTRACT

The tandem tiltwing is one of many aircraft configurations providing vertical takeoff and landing (VTOL). This configuration is expected to be especially suitable for missions requiring VTOL capability combined with high range and space for payload. In this article, an overview of the simulation process of a tandem tiltwing remotely piloted aircraft system (RPAS) without experimental data and its control is given. Contrary to custom, the flight dynamic model, especially the whole aerodynamics model, consists of theoretical equations and interpolations depending on estimated parameters. Compared to complex wind tunnel tests, this approach is less expensive. In order to stabilize the unstable flight characteristics of the tandem tiltwing, a linear-quadratic regulator (LQR) is designed. As the change of operating point of this VTOL aircraft is significant, the LQR has to be gain scheduled. For that, multiple trim points during the transition are ascertained making a controlled transition possible. However, due to the lack of test data, the probability of failure caused by an inaccurate flight controller relying on the flight dynamic model is increased. Hence, a robustness analysis of the closed-loop system is conducted, where the probability of stability of the closed-loop real RPAS is estimated by a Monte Carlo method. For this purpose, all uncertain model parameters are changed based on the normal distribution by defining their standard deviation.

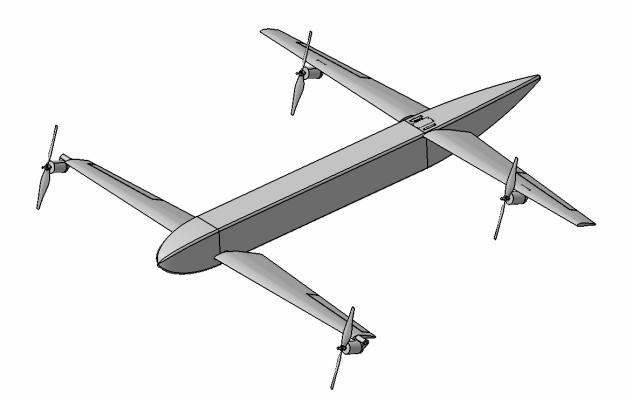


Figure 1: CAD model of the tandem tiltwing

List of Symbols

α	angle of attack
β	sideslip angle
ϑ_1	front wing's tilt angle
ϑ_2	aft wing's tilt angle
ω_i	angular velocity of motor i
$\Phi,\ \Theta,\ \Psi$	Euler angles
b	span
C_D	drag coefficient
C_l	rolling moment coefficient
C_L	lift coefficient
C_m	pitching moment coefficient
C_n	yawing moment coefficient
D	drag
$\boldsymbol{e}$	error vector
L	lift
p	roll rate
P	power
q	pitch rate
Q	lateral force
r	yaw rate
$\boldsymbol{r}$	reference command vector
T	thrust
u	velocity in x direction
$\boldsymbol{u}$	input vector
v	velocity in y direction
w	velocity in z direction
$\boldsymbol{x}$	state vector

List of Indices

η_i	elevon i
ϑ_i	tilt angle i
ω_i	motor i
$b,\ t_i,\ c,\ g,\ w$	coordinate systems

Meaning of Exponents

cm/W	cm relative to the wind (W); example

*Email address: y.beyer@tu-bs.de

1 Introduction

The tandem tiltwing RPAS [1], which is shown in figure 1, is a VTOL aircraft that is controlled like a quadcopter during hover and like a tandem wing during cruise. In similar projects, this configuration is also called quad tiltwing [2, 3, 4] and [5, 6]. It displays some advantages compared to other VTOL airplanes. The span is decreased maintaining the same aspect ratio and consequently the same aerodynamic efficiency resulting in a more compact aircraft. Moreover, the possibility of a relatively large displacement of the center of mass is likewise convenient. Four motors are directly attached to the wings providing a quadcopter configuration in hover mode. Additionally, there are a few advantages compared to a tilt*rotor* configuration, such as the propeller stream is not blocked by the wings and that only two (larger) tilt mechanisms are needed instead of four. Also, a simple trimming possibility is provided by the tiltwings. However, the realization of a tandem tiltwing is relatively complicated due to the two tilt mechanisms carrying the moment of both wings and motors as well as the detached flow on the upper side of the wings during the transition.

Static longitudinal stability of a tandem wing is accomplished by a higher loading of the front wing. However, as a higher loading of the front wing usually decreases efficiency, the loading of both wings should be similar causing static longitudinal instability. Since the motors are used for quadcopter control, they can similarly be used for yaw control during cruise. That makes a rudder as well as vertical stabilizer unnecessary so that they are consequently omitted in order to make the aircraft more lightweight. Thus, the tandem tiltwing displays lateral-directional instability.

The tandem tiltwing can be controlled by the four motors, both tilt angles as well as two elevons. Elevons act as both elevators and ailerons. They are located at the front wing because of higher sensitivity. As the control of the unstable and unsteady flight dynamics of the presented VTOL are very challenging, a high accuracy of the flight dynamics model is crucial in order to properly design an attitude controller. However, no wind tunnel test data is available. Instead, the aerodynamics are calculated by formulas and interpolations covering the complete domain of the angle of attack and the sideslip angle as well as effects depending on the angular rates of the aircraft and wing interaction considering a delayed downwash of the front wing.

Some technical data of the presented tandem tiltwing RPAS is listed in tabular 1.

category	value/product
mass	$5\,\mathrm{kg}$
design airspeed	$20\,\mathrm{m\,s^{-1}}$
measures	$1.4\,\mathrm{m} \times 1.4\,\mathrm{m} \times 0.3\,\mathrm{m}$
propellers	$4\times$ AeroNau 10x5
motors	$4\times$ AXI $2826/10$ (V2)
ESCs	$4\times$ ZTW Spider Series 50A OPTO 2˜6
batteries	$2\times$ Turinigy $5000\,\mathrm{mA\,h}$ 5S 25C LiPo

Table 1: Technical data of the tandem tiltwing RPAS.

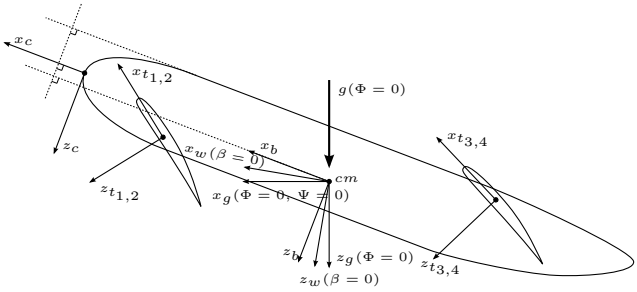


Figure 2: Two-dimensional illustration of the used coordinate systems to simulate the tandem tiltwing. In the general case, x_w, z_w, x_g, z_g and g are three-dimensional in this side view.

2 Flight Dynamic Model

In this section the analytical flight dynamic model of the tandem tiltwing is described. Figure 13 shows an outline of the tandem tiltwing and the needed coordinate systems.

2.1 Motor and Actuator Dynamics

The motor dynamics are based on a first order delay [7] using the motor torque constant, the inertia as well as the internal resistance as parameters. The tilt servos and elevon servos are modeled as second order delays estimating a damping ratio and frequency.

2.2 Propeller Aerodynamics and Dynamics

The propeller thrust and power are calculated with aid of a map provided by [8]. The map depends on the angular velocity as well as the vertical airspeed as it is shown in figure 3 and 4. The following moments and forces are considered [9]:

- thrust vector,
- moment vector due to lever arms,

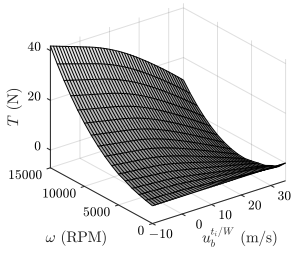


Figure 3: Propeller thrust.

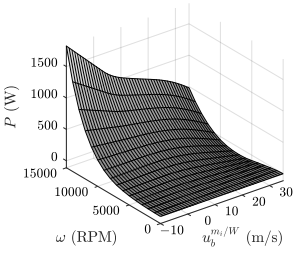


Figure 4: Propeller power.

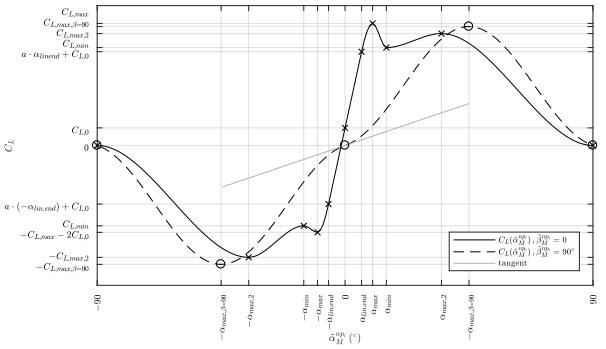


Figure 5: Lift coefficient as a function of the modified angle of attack seen by the tiltwing obtained by interpolation of characteristic points.

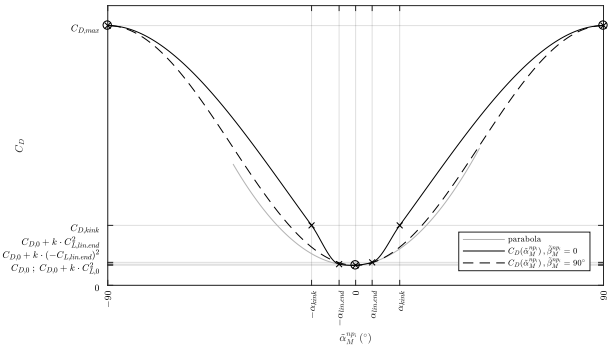


Figure 6: Drag coefficient as a function of the modified angle of attack seen by the tiltwing obtained by interpolation of characteristic points.

- moment vector due to drag,
- moment vector due to inertia when accelerated,
- moment vector due to the propeller gyro effect.

2.3 Fuselage and Wing Aerodynamics

Lift and drag coefficient interpolation. Since the range of the appearing angle of attack is large (it ranges at least from $0°$ to $90°$), the modeling of nonlinear effects is required. The most important aerodynamic coefficients, the lift coefficient C_L and the drag coefficient C_D, must inevitably be defined with respect to the complete range of the angle of attack which is quite uncertain without experimental data. Moreover, at low speed and high wind velocities, the sideslip angle can become large, too. Consequently, the lift coefficient (figure 5) and the drag coefficient (figure 6) must be a nonlinear function with respect to the whole range of the sideslip angle. According to the definition, the angle of attack is defined in the range of $-180° \leq \alpha \leq 180°$ while the sideslip angle is defined in the range of $-90° \leq \beta \leq 90°$. However, working with high angles of attack and sideslip angles a reversed definition of the ranges is desirable. That is why, in this work, a modified angle of attack α_M and a modified sideslip angle β_M are defined changing the order of rotation (appendix 20). For small aerodynamic angles, the difference between the original aerodynamic angles and the modified angles is negligible.

The functions of the lift coefficient as well as the drag coefficient are created by cubic spline interpolation stepwise in between of two points whose slope is known. The points are selected as characteristic points based on experimental data of similar projects [5, 10] as well as literature [11]. Because of the uncertainty of the points in the nonlinear area, all points are defined as parameters which will be changed randomly in the robustness analysis.

Local aerodynamics. There are multiple additional crucial aerodynamic coefficients such as the moment coefficients. Usually, these coefficients are determined by wind tunnel experiments or linear mathematical approaches are used. The linear approaches work with constant aerodynamic derivatives, which are the dimensionless partial derivatives of aerodynamic coefficients with respect to one state, output element or input element. However, for a tiltwing, a linear approach does not seem suitable. That is why a different approach is used.

Except for the lift coefficient, the drag coefficient and the fuselage aerodynamic coefficients, no other aerodynamic coefficients are defined directly. As there are convenient formulas to calculate the fuselage moments, this is done separately. The aerodynamic moments produced by the wings are calculated by the aerodynamic forces acting on movable centers of pressure, thus, acting on lever arms. That means, the aircraft is divided into five parts (left front wing, right front wing, right aft wing, left aft wing and fuselage) whose aerodynamics are calculated independently. Then, the force and moment vector in the cm of the aircraft is calculated by multiplying the local force vectors with the current lever arms to the centers of pressure and summarizing the result including the aerodynamics of the fuselage.

Wing interaction. Until this point all aerodynamic components of the tandem tiltwing were treated independently. Indeed, there is an interaction between the aerodynamic components as well as the propellers. The modeled interaction is restricted to a lift coefficient dependent and delayed downwash of the front wing decreasing the angle of attack of the aft wing according to [11].

2.4 Summary of Assumptions

The complexity of the reality can barely be modeled. The following assumptions of the model may cause substantial deviations of the model and the reality:

- the structure of the aircraft is rigid,
- no interaction between the propellers, the wings and the fuselage except of a simple downwash model,

- no noncontinuous phenomenons e.g. for the lift coefficient or the actuators,
- no dependency of the aerodynamics on the Reynolds number and Mach number,
- the thrust does not depend on the lateral velocity seen from the propeller.

3 Stability and Control

In this section, the stability of the tandem tiltwing is analyzed by means of the indirectly defined aerodynamic derivatives as well as root loci. For these linear methods, the model has to be linearized in the first place. Based on the obtained linear models, a gain scheduled LQR is designed.

3.1 Operating Points and Linearization

As the desired control strategy requires a linear model, the flight dynamic model has to be linearized in multiple operating points. At first, these operating points have to be found starting with steady-state operating points. Therefore, all steady states are defined and if known, the value also is defined. This is done by changing the front tilt angle in the following steps:

$$\begin{aligned}\vartheta_1(^\circ) = (\ & 90 \quad 80 \quad 70 \quad 60 \quad 50 \quad 40 \quad 30 \quad 20 \\ & 14 \quad 10 \quad 7 \quad 3\ )\ .\end{aligned} \tag{1}$$

The outcome is the input vector as well as the airspeed. In the next step, for all obtained airspeeds an additional trimming for a forward acceleration of $\mathrm{d}u_b^{cm/W}/\mathrm{d}t = 3\,\mathrm{m\,s^{-2}}$ as well as a deceleration of $\mathrm{d}u_b^{cm/W}/\mathrm{d}t = -3\,\mathrm{m\,s^{-2}}$ is conducted. While the forward acceleration is easy to trim, the demanded deceleration can not be achieved in horizontal flight. That is why a vertical speed is allowed for the deceleration in order to convert kinetic energy to potential energy. The outcome of the unsteady trimming is the input vector.

3.2 Open-Loop Stability

The static stability calculations can be split into longitudinal static stability and lateral-directional static stability.

Longitudinal static stability. A negative slope of the pitching moment coefficient with respect to the angle of attack is the most important condition for longitudinal static stability. The corresponding illustration is shown in figure 7 varying the relative center of mass (cm) position $0 \leq h \leq 1$ which is zero at the neutral point of the front wing and one at the neutral point of the aft wing. According to the figure, the longitudinal motion of the tandem tiltwing becomes unstable if $h \geq 0.46$. However, because of efficiency purposes $h = 0.5$, which causes static instability, is desired. In figure 8, the pitch damping is illustrated. Small angles of attack lead to a large pitch damping, while the positive slope of the curve corresponding to $\vartheta_1 = 14^\circ$ causes unstable pitch damping.

Lateral-directional static stability. The conditions of lateral-directional static stability mainly are a negative slope of the rolling moment coefficient (figure 9) and a positive slope of the yawing moment coefficient (figure 10) with respect to the sideslip angle. However, the yawing moment is unstable because the unstable moment produced by the fuselage is not overcompensated by a vertical stabilizer. The other stability derivatives depending on the rates were investigated equally.

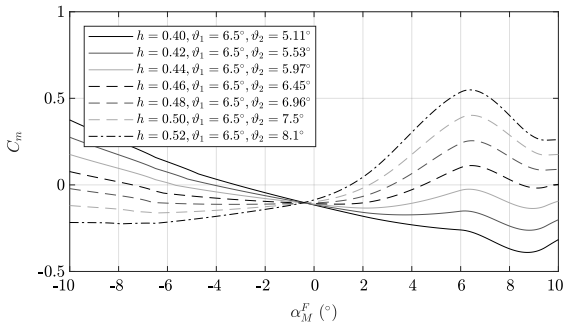


Figure 7: Longitudinal static stability analysis regarding the slope of the pitching moment coefficient C_m with respect to the angle of attack α for different center of gravity x-positions.

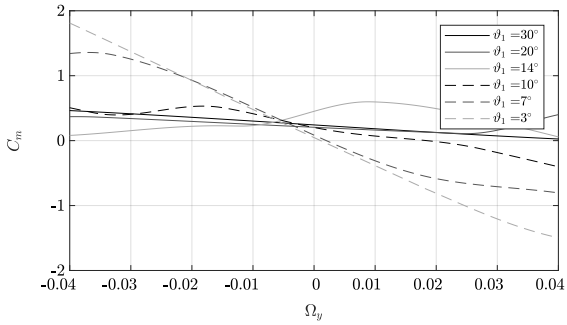


Figure 8: Longitudinal static stability analysis regarding the slope of the pitching moment coefficient C_m with respect to the dimensionless pitch rate $\Omega_y = q \cdot \bar{c}/V^A$ for different tilt angles ϑ_1 in trimmed steady horizontal flight.

Dynamic stability. While static stability describes the tendency of a system to return to the trimmed state after being perturbed, dynamic stability contains the behavior of the system over time. For the dynamic stability analysis, two root loci are used. In figure 11 the dependency of the poles to the relative cm position h is illustrated. Similarly to the longitudinal static stability analysis, according to the poles, the longitudinal motion becomes asymptotically unstable if $h \geq 0.46$ while the lateral-directional motion is unstable permanently. Moreover, figure 12 shows the poles of the tandem tiltwing during all steady-state operating points. The dutch roll motion (lateral-directional motion) is unstable once a moderate airspeed is exceeded because of the lack of a vertical stabi-

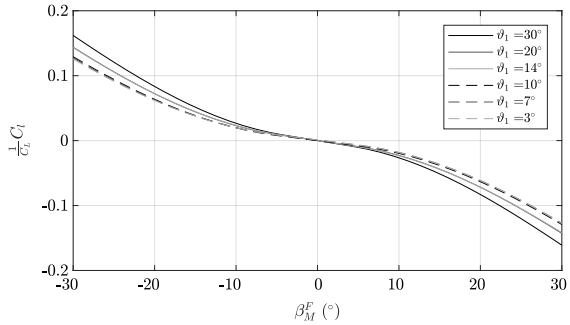


Figure 9: Lateral-directional static stability analysis regarding the slope of the rolling moment coefficient C_l with respect to the sideslip angle β for different angles of attack α.

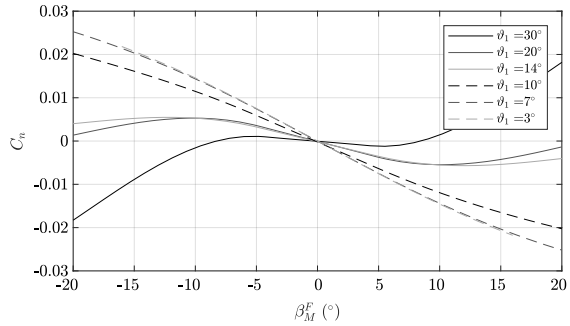


Figure 10: Lateral-directional static stability analysis regarding the slope of the yawing moment coefficient C_n with respect to the sideslip angle β for different angles of attack.

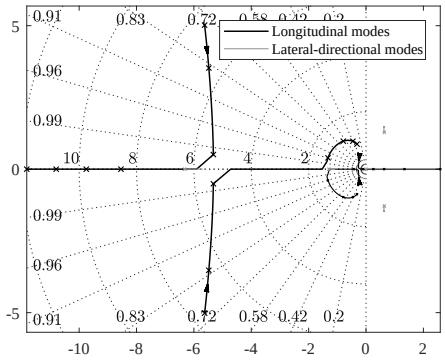


Figure 11: Stability analysis of the design point by means of a root locus varying the relative center of mass position h from $0.40 < h < 0.52$, crosses in equivalent intervals.

lizer. Regarding the longitudinal motion, the short period mode is asymptotically unstable all the time. Both longitudinal and lateral-directional motion are the most unstable for a tilt angle of $\vartheta_1 = 14°$.

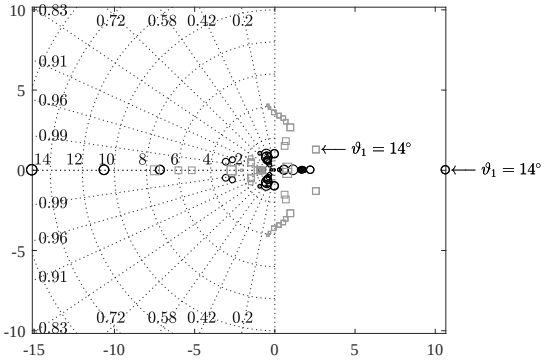


Figure 12: Stability analysis by means of a pole plot varying the steady-state operating point.

○ longitudinal motion,
□ lateral motion,
smallest markers: hover mode,
largest markers: cruise.

3.3 Control structure

The control structure, which is illustrated in figure 13, contains a gain scheduled (GS) trim command acting as feedforward as well as a GS linear quadratic regulator (LQR).

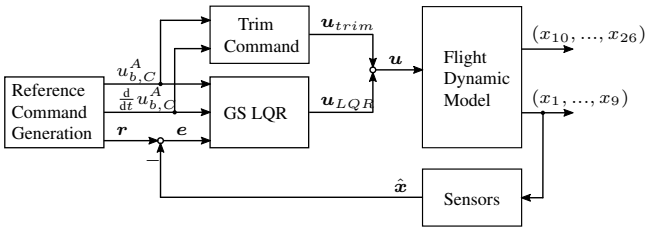


Figure 13: Diagram of the gain scheduled (GS) flight control structure.

The trim command depends on the desired airspeed as well as the desired acceleration, where the commanded acceleration is always limited to $-3\,\mathrm{m\,s^{-2}} \leq \mathrm{d}u_b^{cm/W}/\mathrm{d}t \leq 3\,\mathrm{m\,s^{-2}}$. The trim commanded tilt angles as well as the motor commands, which are normalized to $0 \leq u_{\omega_i} \leq 1$, are shown in figure 14 and 15. The trim command for the elevon deflection is always the neutral deflection.

3.4 LQR Design

According to [12] the weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ for the LQR design can be chosen by determining the maximum allowed states respectively control inputs. Since the LQR is gain scheduled, the weighting matrices must be chosen sev-

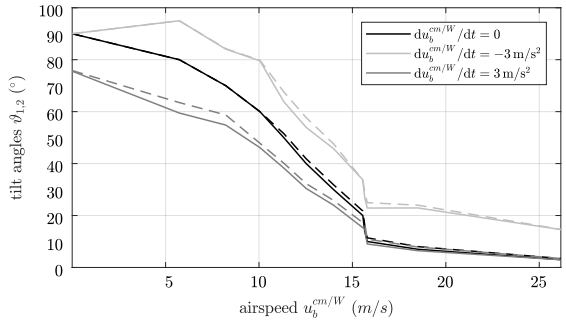


Figure 14: Tilt angle trimming (feed-forward) of the front wing (——) and the aft wing (- - -) depending on the airspeed as well as on the forward acceleration.

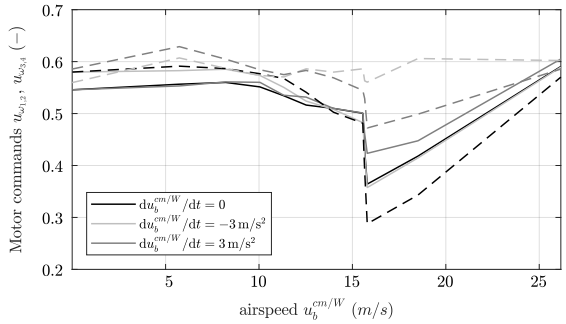


Figure 15: Motors command trimming (feed-forward) of the front motors (——) and the aft motors (- - -) depending on the airspeed as well as on the forward acceleration.

eral times (depending on the tilt angle of the front wing):

$$\boldsymbol{Q}(\vartheta_1) = \text{diag}\left(\frac{1}{p_{max}^2}, \frac{1}{q_{max}^2}, \frac{1}{r_{max}^2}, \frac{1}{\Phi_{max}^2}, \frac{1}{\Theta_{max}^2}, \frac{1}{\Psi_{max}^2}, \frac{1}{{u_{b,max}^{cm/W}}^2}, \frac{1}{{v_{b,max}^{cm/e}}^2}, \frac{1}{{w_{b,max}^{cm/e}}^2}\right), \tag{2}$$

$$\boldsymbol{R}(\vartheta_1) = \text{diag}\left(\frac{1}{u_{\eta_1}^2}, \frac{1}{u_{\eta_2}^2}, \frac{1}{u_{\vartheta_1}^2}, \frac{1}{u_{\vartheta_2}^2}, \frac{1}{u_{\omega_1}^2}, \frac{1}{u_{\omega_2}^2}, \frac{1}{u_{\omega_3}^2}, \frac{1}{u_{\omega_4}^2}\right). \tag{3}$$

The weighting matrices are defined for four trimpoints: $\vartheta_1 = 90°$, $\vartheta_1 = 80°$, $\vartheta_1 = 10°$, $\vartheta_1 = 7° \wedge \vartheta_1 = 3°$. Then, they are linearly interpolated in between of these points, where the weighting matrices for both cruising operating points are the same.

3.5 Closed-Loop Stability

An LQR always stabilizes the ideal linear system. Figure 16 illustrates the poles of the open-loop and the closed-loop system in its design point. It shows that the modes of the closed-loop model are well damped. However, the frequency of oscillation is increased which could increase the risk of causing structural oscillations.

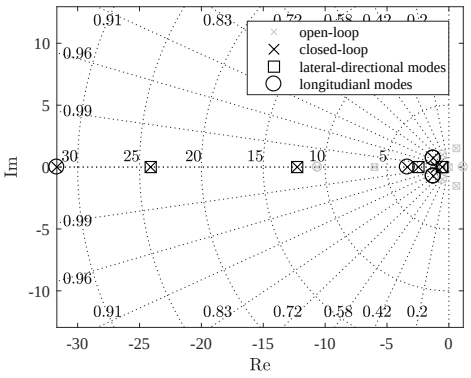


Figure 16: Pole plot of the steady-state trim point corresponding to the tilt angle $\vartheta_1 = 7°$.

3.6 Monte Carlo Robustness Analysis

The flight dynamic model contains about 100 parameters. As these parameters are more or less uncertain, the robustness of the system is investigated by varying the parameters maintaining the same controller gains. Therefore, these parameters are normally distributed by defining the expected value and the standard deviation. Then, all parameters are combined several times in order to determine the random closed-loop system dynamics. The desired outcome is the probability of failure of the real system. The necessary number of parameter combinations strongly depends on the actual probability of failure as well as the desired confidence interval [13, 14, 15].

The system stability is investigated by means of a pole distribution treating the system as a linear time-invariant system [15]. Since the tandem tiltwing model is significantly nonlinear and time-variant, the system stability is additionally investigated with aid of a dynamic maneuver covering a whole transition both forward and backwards.

Figure 17 shows the poles of 800 parameter combinations of the most unstable open-loop operating point of the tandem tiltwing and figure 18 shows the same plot for the design point. While the poles of the design point remain always stable, the most unstable open-loop operating point stays unstable in several cases. This can be explained by the high non-linearity in this area. In this operating point, the onset and ending of the wing's stall occurs (figure 9).

According to the linear-time invariant analysis, for most parameter sets, there is at least one operating point which is unstable. Mostly entirely, the unstable operating points occur at the end of the forward transition respectively at the beginning of the backwards transition because of the uncertainty of the ending and onset of the stall.

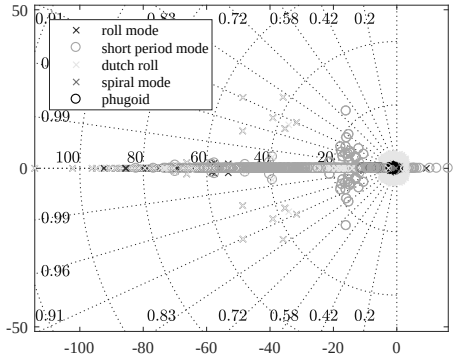


Figure 17: Linear time-invariant robustness analysis by means of a pole map of 800 parameter combinations regarding the decelerated operating point corresponding to the tilt angle of $\vartheta_1 = 14°$.

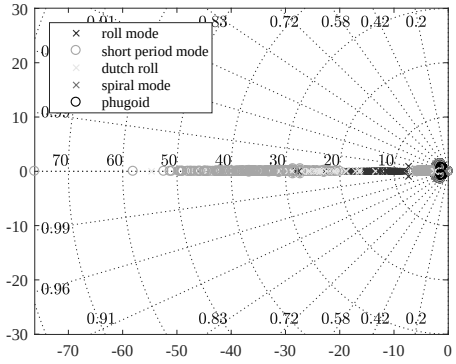


Figure 18: Linear time-invariant robustness analysis by means of a pole map of 800 parameter combinations regarding the steady-state operating point corresponding to the tilt angle of $\vartheta_1 = 7°$.

However, the dynamic maneuver robustness analysis shows that these unstable operating points are very brief. Thus, in most cases they do not lead to a failure of the system. According to the dynamic maneuver, the probability of failure can be expected to be less than $15\,\%$ within a confidence interval of $95\,\%$. All simulated failures occurred at the end of the forward transition.

4 Flight Test

In order to validate the approach of this work, a prototype of the tandem tiltwing, called Changyucopter, was constructed (figure 19).

The designed controller was implemented on a Pixhawk flight controller [16]. This flight controller contains sensors like accelerometers, gyroscopes, compasses, an airspeed sensor and a GPS receiver. The open source firmware for the Pixhawk flight controller (PX4 or ArduPilot) runs an extended Kalman filter (EKF) as state estimator. Thus, this EKF can be used for the LQR with state feedback.

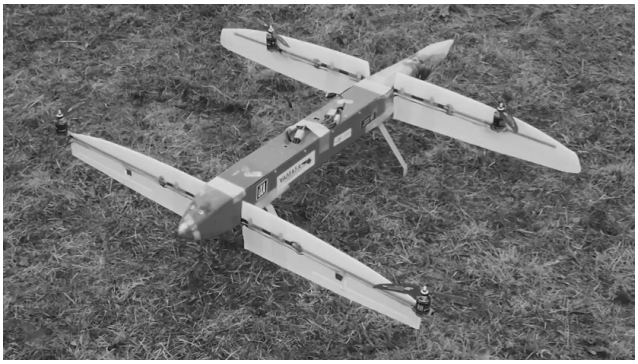


Figure 19: The first prototype of the Changyucopter tandem tiltwing.

The planned flight tests are supposed to show if the described flight dynamic model without experimental data is valid. However, possible errors of the prototype could not only be caused due to the lack of experimental data, but also due to the made assumptions. Especially the neglecting of a flexible structure and most interactions between the aerodynamic components are expected to be critical.

5 Conclusion

The aerodynamics of a tandem tiltwing RPAS are modeled without experimental data. Therefore, the lift and drag coefficient are interpolated with respect to the aerodynamic coefficient based on estimated characteristic points. Multiple additional aerodynamic coefficients are defined indirectly by defining moving centers of pressure. For the stability analysis and the control design, multiple steady-state, accelerated and decelerated trim points are sought and linearized. The stability analysis of the open-loop system shows that the stability derivatives are reasonable. As both the longitudinal and the lateral-directional motion are unstable, an active stabilization by a controller is crucial. The control structure consists on a gain scheduled feed-forward trim command as well as a gain scheduled LQR stabilizing the system. Since the model relies on many uncertain parameters, a Monte Carlo robustness analysis of the closed-loop system is conducted. In most trim points the system remains stable, however, there is a critical area within the transition. Instability issues occur when the tiltwing's stall starts or end because in that area the aerodynamics are highly nonlinear. Nevertheless, the instability only occurs for a short moment and the dynamic simulation shows that the RPAS quickly recovers. The flight dynamic model and the designed controller are going to be tested with a prototype of the tandem tiltwing.

REFERENCES

[1] Yannic Beyer. Flight control design and simulation of a tandem tilt wing RPAS. master thesis, TU Braunschweig, Institute of Flight Guidance, 2017. unpublished.

[2] Kaan Taha Öner, Ertuğrul Çetinsoy, Mustafa Ünel, Mahmut Faruk Akşit, Ilyas Kandemir, and Kayhan Gülez. Dynamic model and control of a new quadrotor unmanned aerial vehicle with tilt-wing mechanism. 2008.

[3] E. Cetinsoy, S. Dikyar, C. Hancer, K.T. Oner, E. Sirimoglu, M. Unel, and M.F. Aksit. Design and construction of a novel quad tilt-wing UAV. *Mechatronics*, 22(6):723–745, September 2012.

[4] Kaan Taha Öner, Ertuğrul Çetinsoy, EFE SIRIMOĞLU, Cevdet Hançer, Mustafa Ünel, Mahmut Faruk Akşit, Kayhan Gülez, and Ilyas Kandemir. Mathematical modeling and vertical flight control of a tilt-wing UAV. *Turkish Journal of Electrical Engineering & Computer Sciences*, 20(1):149–157, 2012.

[5] Koji Muraoka, Noriaki Okada, and Daisuke Kubo. Quad tilt wing VTOL UAV: Aerodynamic characteristics and prototype flight. In *AIAA Infotech@ Aerospace Conference and AIAA Unmanned... Unlimited Conference*, page 1834, 2009.

[6] Koji Muraoka, Noriaki Okada, Daisuke Kubo, and Masayuki Sato. Transition flight of quad tilt wing VTOL UAV. In *28th Congress of the International Council of the Aeronautical Sciences*, pages 2012–11, 2012.

[7] Samir Bouabdallah, Andre Noth, and Roland Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference On*, volume 3, pages 2451–2456. IEEE, 2004.

[8] Landing Products Inc. APC Propeller Performance Data. Website, 2017. URL: `https://www.apcprop.com/Articles.asp?ID=270` [accessed on May 6, 2017].

[9] Samir Bouabdallah. *Design and Control of Quadrotors with Application to Autonomous Flying*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 2007.

[10] Paul M. Rothhaar, Patrick C. Murphy, Barton J. Bacon, Irene M. Gregory, Jared A. Grauer, Ronald C. Busan, and Mark A. Croom. NASA Langley Distributed Propulsion VTOL TiltWing Aircraft Testing, Modeling, Simulation, Control, and Flight Test Development. American Institute of Aeronautics and Astronautics, June 2014.

[11] Hermann Schlichting and Erich Truckenbrodt. *Aerodynamik des Flugzeuges - Teil 2*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[12] Brian L. Stevens, Frank L. Lewis, and Eric N. Johnson. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. John Wiley & Sons, Hoboken, N.J, third edition edition, 2016. OCLC: ocn935444384.

[13] Robert F. Stengel and Laura R. Ray. Application of stochastic robustness to aircraft control systems. *Journal of Guidance, Control, and Dynamics*, 14(6):1251–1259, November 1991.

[14] Robert F. Stengel and L. E. Ryan. Stochastic robustness of linear time-invariant control systems. *IEEE Transactions on Automatic Control*, 36(1):82–87, 1991.

[15] Laura Ryan Ray and Robert F. Stengel. A Monte Carlo approach to the analysis of control system robustness. *Automatica*, 29(1):229–236, 1993.

[16] Lorenz Meier. Pixhawk. Website, 2017. web: `https://pixhawk.org` [accessed on September 01 2017].

APPENDIX A: MODIFIED AERODYNAMIC ANGLES

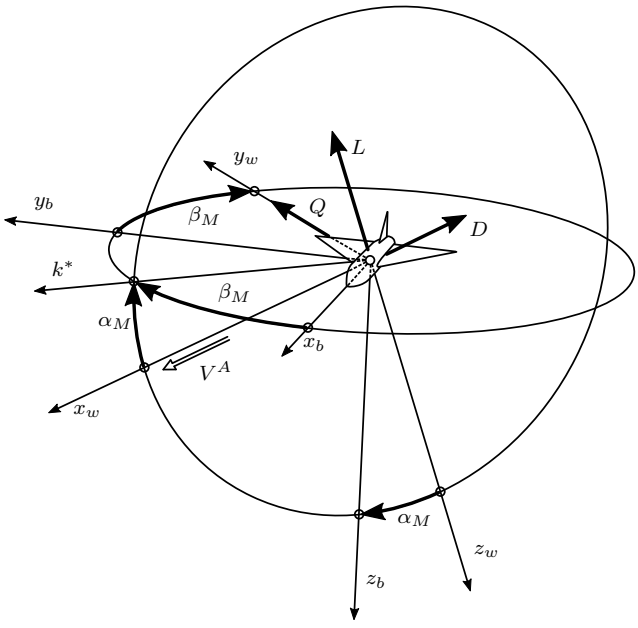


Figure 20: Definition of the modified angles α_M and β_M and the aerodynamic quantities.

Using MAVs for Atmospheric Wind Measurements: Opportunities and Challenges

Simon Watkins1*, Mohamed Abdulghani, Sam Prudden, Mathew Marino, Reece Clothier, Alex Fisher and Ashim Panta
RMIT University Australia

ABSTRACT

MAVs are increasingly being used as flying sensors due to their ability to be positioned in hard to access locations for relatively low risk and cost. The use of fixed wing and multi-rotor MAVs as flying anemometers were investigated by instrumenting one of each with multi-hole pressure probes (simplified versions of the TFI Cobra probes) and obtaining data on windy days. A batch of probe heads were produced by stereolithography (STL) rapid prototyping and it was found that a universal calibration could be used for one batch unless extreme levels of accuracy are required. The craft, on-board sensors and the signal processing techniques are described. The limited endurance of multirotors was found to be a significant limitation when trying to obtain the relatively long samples of data (required for good descriptions of atmospheric turbulence). As wind data are usually required at several spatial locations (e.g. vertically displaced replicating the function of several anemometers on a mast) it is necessary to have several craft flying concurrently. Another challenging aspect was holding a steady position for the multi rotor craft. Automated position holding is part of on-going work and we plan to investigate using the technique of stabilising the craft by upstream flow measurements (proven on the fixed wing MAV) for multi-rotors.

1 INTRODUCTION AND OBJECTIVES

MAVs are increasingly being used as flying sensors due to their ability to be positioned in hard-to-access locations for relatively low risk and cost. One potential application is documenting the atmospheric wind, including its interaction with structures and terrain, using either fixed wing and/or multi-rotor craft. The advantage of multi-rotors is that they can hover (either with respect to the Earth or wind), whereas fixed-wing generally cannot. However fixed-wing MAVs can fly faster which can be an advantage if a long time record of the wind needs to be captured (i.e. wind run). As atmospheric flows vary continuously with time and space it is often required to document the turbulence characteristics, such as intensities, scales and spectra in three orthogonal directions. For meteorological and wind engineering studies sampling times are at least 20 minutes and often over an hour and sampling frequency can be relatively low (O(1 to 10 Hz). Of relevance to wind effects on buildings structures, cars, aircraft etc. is the turbulent energy in the micro meteorological range (generally known as gusts), as opposed to weather map fluctuations driven by larger scale climatic events. The duration of the sample lengths fall in the spectral gap centred about one hour. For more details see [1].

Typically MAVs are less than a metre wingspan and can be of either fixed wing or multi rotor configuration. The advantage of multi rotors over fixed wing craft is that they can hover (either with respect to the Earth or the atmospheric wind), take off and land vertically and are more manoeuvrable (particularly in the lateral and vertical directions and in yaw). Fixed-wing aircraft generally can fly faster and for longer, which can be an advantage if a long time record of the wind needs to be captured (i.e. wind run) and/or when strong winds are to be measured. Flight speeds of micro fixed wing aircraft are from 5 to 30 m/s, with a similar top speed for multi rotor craft.

Traditionally wind data has been obtained from fixed locations, using mast-mounted anemometers, Lidar etc., where the sensor is essentially rigidly mounted. Thus these methods are not significantly affected by motion of the support. This is not the case for airborne measurements unless the aircraft is held steady relative to the ground. When the sensor is moving; either in a steady translation, or perturbed by motion of the aircraft, the ensuing spurious motions can sometimes be removed, thus the resulting data is then in the (Earth) body axis system. However MAVs are small, have low mass and MOI, and fly at relatively low speed thus perturbations from turbulence can be significant. This is particularly noticeable for fixed wing MAVs in the roll axis ([2]). As MAVs generally carry inertial measurement systems (often using the data in feedback control to assist in flight stabilisation) the acceleration data can be integrated to give velocity information about sensor movement which can be then removed from the wind data.

Our objectives are to understand the opportunities and challenges of measuring wind utilising a fixed wing and a

*Email address(es): simon@rmit.edu.au

quad-rotor aircraft. The sensors used were multi-hole probes based on the TFI Cobra probe [1].

2 Instrumentation: MAVs and Sensors

2.1 Fixed Wing

A Slick 360 aerobatic aircraft was used which is neutrally stable, has a 0.490 m span NACA0012 aerofoil section wing, a typical flying mass of 130g mass and was fabricated via injection moulding from relatively rigid high-density foam. The wings had no twist or dihedral and the large ailerons (30% x/c) extended over the entire semi-span. The reason for selecting a model with nominally neutral stability was that with suitable active stabilisation systems a relatively straight and level flight path could be held even under turbulent conditions. Initially a standard inertial-based stabilisation was used, but then an improved patented stabilisation system was fitted. This was found to significantly improve the steadiness of flight in the turbulent atmosphere and utilised upstream measurements of pitch angle perturbations about to impinge on the wing to give increased time for the control system to reduce the unwanted roll reactions. The improved stabilisation system utilised two carbon fibre multi-hole pressure probes (described later). The model, with probes fitted, can be seen in Figure 1 and details of the method can be found in [3].



Figure 1: Fixed Wing Aircraft

2.2 Multi Rotor

A bespoke racing quadcopter was modified to incorporate multi-hole pressure probes as shown in Figure 3. Under the main probe shaft can be seen a carbon-fibre protection strut which is to provide protection during take-off and landing. The placement of the probes relative to the rotor flow field was part of a separate research program [4, 5] and for the work reported here the probe head was sufficiently far removed from the rotor flow field for the influence to be insignificant.

[1] http://www.turbulentflow.com.au/Products/CobraProbe/CobraProbe.php

2.3 Wind Sensing Instrumentation

Three-component velocity sensing was via multi-hole pressure probes, which are being increasingly used for turbulent flow documentation as they are less fragile and expensive than hot-wire or laser-Doppler anemometers. An overview of the technique can be found in [6]. Typically such probe systems are individually calibrated; both in terms of angle and frequency response. A dynamic calibration is often applied to correct for the amplitude and phase changes that occur due to the tubing system and pressure transducer response. The data zone of acceptance for the four-hole head design used here (see Figure 3) is a 90 degree cone. Data that lie outsize this zone can be rejected.

For commercially available multi-hole probe systems, data acquisition and processing is normally performed via a personal or laptop computer. For the current applications it was desirable to incorporate on-board data logging, with data downloaded and analysed post-flight. Thus the probes and associated hardware had to be lighter than existing commercial systems. As the risk of ground or building impact was considered high, low cost, easily replaceable probe heads were necessary. A series of heads were manufactured via rapid prototyping, including an assessment of whether a single, (or average calibration) could be used for a batch of probe heads. This work is described in [4]. A bespoke calibration was used for the two flying systems described here. The outputs of the probe were connected to four pressure transducers with temperature compensation and analog-to-digital conversion. The digital pressure data was then transmitted to a central FPGA-based processing unit, which also interfaced with a GPS system and IMU. All data was logged to a microSD card. This setup allowed for very accurate (O(10us)) synchronization between IMU/GPS and pressure data, which is necessary to correct for any movement of the aircraft, and resolve the measured wind velocities into the global frame.

3 Wind Measurements

Several flights were undertaken using the fixed wing aircraft including outdoors and in three wind engineering tunnels. The outdoor flight path depended upon the wind speed; when winds of higher magnitude than the flight speed were encountered the aircraft could be held closely stationary (relative to the Earth), whereas for lower winds the aircraft was moving relative to the ground thus obtained a wind run which could (depending upon terrain/fetch) be hundreds of meters long. Figure 4 shows a raw pressure data from the probes with velocity spectra shown in Figures 5 and 6. The typical error in pressure measurements was ± 1 Pa and the



Figure 2: Rotorcraft

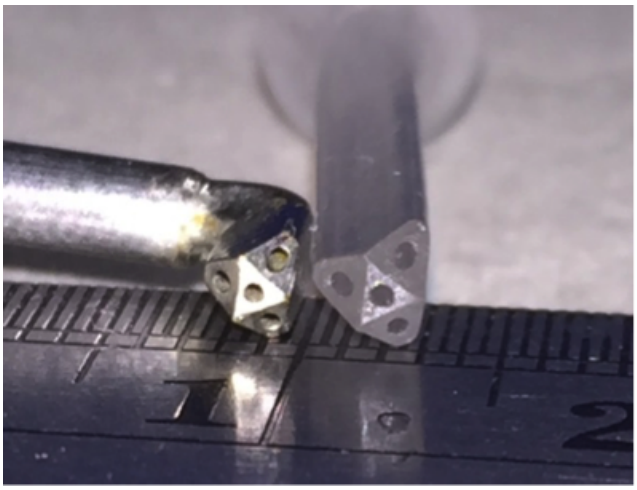


Figure 3: TFI Cobra Probe next to the STL-manufactured Probe

spectra show close alignment with the theorised Von Karman Spectra.

To date only one short trial flight was conducted with the quadrotor. There were issues identified with holding a steady position relative to the Earth but the trial demonstrated the potential of using a quadrotor- mounted system, see Figure 7,8 for a plot of elevation vs time (from GPS) and wind longitudinal velocity vs elevation. Estimated error in elevation measurements was $\pm$2m.

4 Concluding Thoughts and Lessons Learnt

The endurance of rotor craft generally means that sample times will be limited to below 20 minutes and in our initial flight trial this was approximately 12 minutes. This is an issue especially if the wind is to be measured at significant elevations (of order of hundreds of metres) as considerable battery energy is utilised to climb and descend to the desired

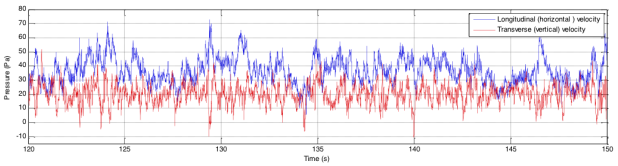


Figure 4: Airborne raw pressure measurements

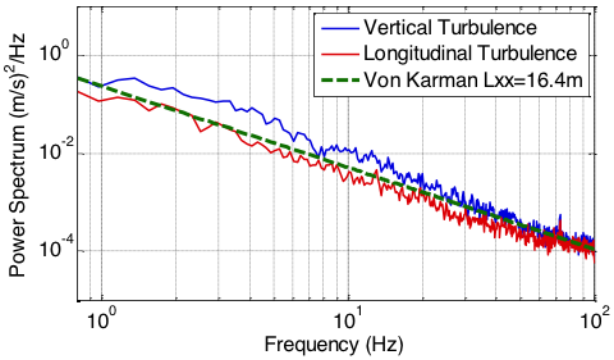


Figure 5: Turbulence velocity spectrum outdoors @ Ti=8%

altitude leaving little for the measurement phase. As our trial was limited to one quadrotor we attempted to obtain data at several elevations to document the velocity profile but clearly the sample length was not long enough to be statistically useful for meteorological and wind engineering purposes. However if only the high frequency end of the spectra is required, which will be the case for turbulence studies on MAVs, this should not pose a problem.

Flying rapidly through the turbulence field can be advantageous for both types of craft since a long wind run can be obtained, effectively compressing the sampling time.

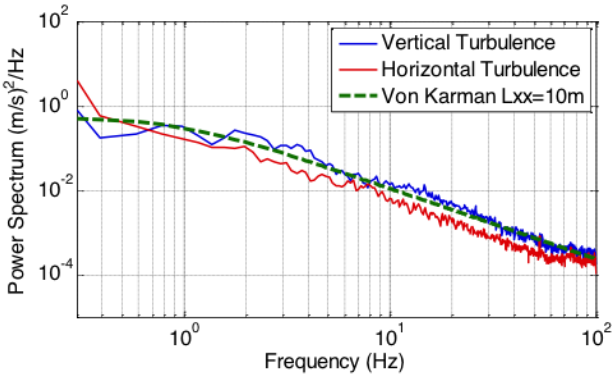


Figure 6: Turbulence velocity spectrum in Wind tunnel @ Ti= 7.5%

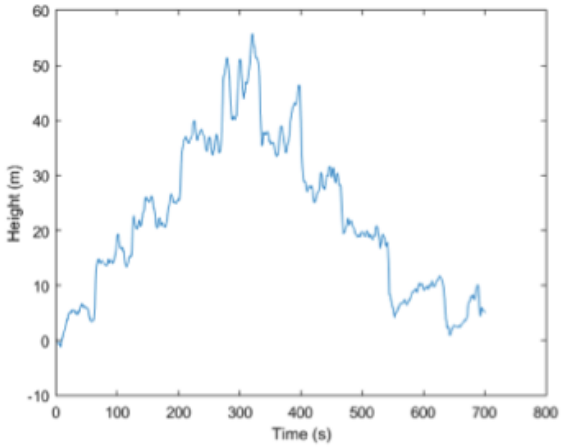


Figure 7: Elevation vs Time from Initial Flight of Quadrotor

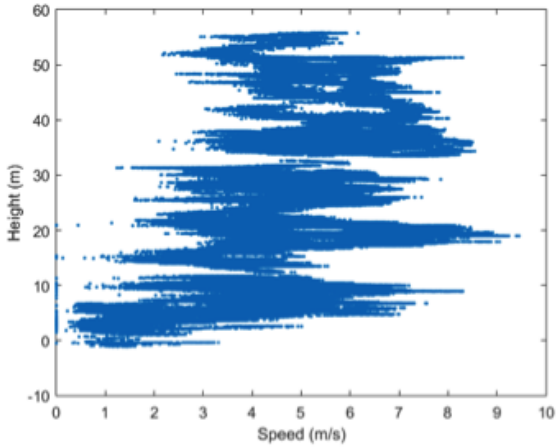


Figure 8: Velocity Profile from Initial Flight of Quadrotor

Clearly this depends upon the terrain/fetch in question, but where this is permissible it is a useful technique. As long as the sampling frequency and system response is sufficiently high, data samples can then be uncompressed (by removing the mean flight speed of the aircraft) which gives a longer effective time record of data. This can technique can be beneficial under low wind conditions when (from a stationary sampling location) very large flow angles are encountered; greater than the 90 degree zone of acceptance for the four hole probes used here. Increased flight speed reduces the magnitude of the fluctuating angles. Data can then be post processed to remove the effect of flight speed in a similar manner to uncompressing the velocity magnitudes. Note that this technique has been applied to hot wires which have been flown through bluff body wakes enabling very large fluctuating angles to be captured. ([8]).

The multi-hole pressure probes were found to be very suited to taking 3-D wind turbulence data and rapid prototyping can mass produce probes reasonable well. With typical STL techniques one universal calibration can be used for a batch, unless extreme levels of accuracy are required. Being a pressure-based velocity measuring system it has a nominally parabolic response of output to velocity. This makes them insensitive at low wind speeds but as data are usually required for high wind speed (¿5m/s) this was not a problem.

The technique of measuring the upstream flow on the fixed wing craft and utilising the information for stabilisation control was found to be very useful; particularly in roll. It enabled flight in very high frequency energetic turbulence including in wind engineering wind tunnels with levels of turbulence of over 20% and of a scale of similar dimension to the craft.

5 Where To From Here?

As wind data are usually required at several spatial locations (e.g. vertically displaced replicating the function of several anemometers on a mast) it is necessary to have several craft flying concurrently. The lack of endurance is problematic but swarming will be useful. We plan to extend to a swarm of four rotor craft.

Another challenging aspect was holding a steady position for the multi rotor craft including at several elevations as shown in Figure 7. Automated position holding is part of on-going work and we plan to investigate using the technique of stabilising the craft by upstream flow measurements (proven on the fixed wing MAV) for multi-rotors.

It is envisaged that the continuing reduction in cost for automated micro planes will lead to them being used increasingly in swarms, with commensurate reduction in the use of mast-mounted anemometers.

6 Acknowledgments

Much of the work on the fixed wing flight stabilisation system has been funded via USAF grants administered under the Asian Office of Aerospace Research and Development (AOARD 05-75) and a funding from the Australian Defence Science Institute (DSI-0200314379). Sam Prudden and Ashim Panta are funded via Australian Federal Government APA doctoral scholarship.

References

[1] S Watkins, M Thompson, B Loxton, and M Abdulrahim. On low altitude flight through the atmospheric boundary layer. *International Journal of Micro Air Vehicles*, 2(2):55–68, 2010.

[2] S Watkins, J Milbank, Benjamin J Loxton, and W.H Melbourne. Atmospheric winds and their implications for microair vehicles. *AIAA journal*, 44(11):2591–2600, 2006.

[3] A Mohamed, M Abdulrahim, S Watkins, and R Clothier. Development and flight testing of a turbulence mitigation system for micro air vehicles. *Journal of Field Robotics*, 33(5):639–660, 2015.

[4] S Prudden, A Fisher, A Mohamed, and S Watkins. An anemometer for uas-based atmospheric wind measurements. In *17th Australian International Aerospace Congress: AIAC 2017, Melbourne Australia*, pages 303–308. Engineers Australia, Royal Aeronautical Society, 26-29 Feb, 2017.

[5] S Prudden, A Fisher, A Mohamed, and Watkins. A flying anemometer quadcopter: Part 1. In *International Micro Air Vehicle Conference: IMAV 2016, Beijing China*, 2016.

[6] S Watkins, P.D Mousley, and J.D Hooper. Measurement of fluctuating flows using multi-hole probes. In *9th International Congress on Sound and Vibration, Orlanda Florida USA*. International Institute of Acoustics and Vibration (IIAV), 8-11 July, 2002.

[7] S Watkins, P.D Mousley, and G Vino. The development and use of dynamic pressure probes with extended cones of acceptance (eca). In *15th Australasian Fluid Mechanics Conference, Sydney Australia*.

[8] J.H Watmuff, A.E Perry, and M.S Chong. A flying hot-wire system. *Experiments in Fluids*, 1(2):63–71, 1983.

[9] M Marino, A Fisher, R Clothier, S Watkins, S Prudden, and C.S Leung. An evaluation of multi-rotor unmanned aircraft as flying wind sensors. *International Journal of Micro Air Vehicles*, 7(3):285–299, 2015.

[10] A. Mohamed, S. Watkins, and R.A. Clothier. Methods and systems for attenuating the effects of turbulence on aircraft, dec 2015. WO Patent App. PCT/AU2015/000326.

Bio-inspired Wind Field Estimation-Part 1: AoA Measurements Through Surface Pressure Distribution

Nikola Gavrilović*
Institut Supérieur de l'Aéronautique et de l'Espace-SUPAERO, Toulouse, 31400, France
Murat Bronz‡
l'Ecole National de l'Aviation Civile-ENAC, Toulouse, 31055, France
Jean-Marc Moschetta‡, Emmanuel Bénard⊥and Philippe Pastor△
Institut Supérieur de l'Aéronautique et de l'Espace-Supaéro, Toulouse, 31400, France

ABSTRACT

One of the major challenges of Mini-UAV flight is unsteady interaction with turbulent environment while flying in lower levels of atmospheric boundary layer. Following inspiration from nature we expose a new system for angle of attack estimation based on pressure measurements on the wing. Such an equipment can be used for real-time estimation of the angle of attack during flight or even further building of wind velocity vector with additional equipment. Those information can find purpose in control and stabilization of the aircraft due to inequalities seen by the wing or even for various soaring strategies that rely on active control for energy extraction. In that purpose flying wing UAV has been used with totally four span-wise locations for local angle of attack estimation. In-flight angle of attack estimation of differential pressure measurements have been compared with magnetic sensor with wind vane. Difference in local angle of attack at four span-wise locations has confirmed spatial variation of turbulence. Moreover, theoretical energy dissipation of wind fluctuations described by Kaimal spectrum has shown acceptable match with measured ones.

1 INTRODUCTION

There are strong indications that birds use their feathers for sensing of flow perturbations over their wingspan. Being fluffy and subjected to fluttering provoked by small disturbances, birds have natural sensory system which enables them to "feel" flow disorders along wing. Another convenience of their elastic body structure is capability of using adaptronics for various turbulent flight regimes. Eventual immediate action due to surface pressure fluctuations by modifying wing geometry or profile curvature allows quick and effective response in gusty environment.

However, for a variety of reasons, it is understood that identical copies from nature to man-made technologies are not feasible. Instead, a creative inspiration and conversion into technology is often based on various steps of abstraction.

Mini UAVs usually fly at lower levels of atmospheric boundary layer where the turbulence intensity is significantly increased due to the proximity of the ground. Such a complex surrounding implies intricate interaction between terrains geometry, physical conditions and varying meteorology. Being three-dimensional, turbulence scales larger than wingspan would result in only pitching attitude of the aircraft. In contrary, case of turbulence smaller than wingspan leads to unequal lift distribution and need for control of oncoming roll and yaw moments. These information can be of particular interest in case of gust energy extraction flight strategy.

The performance of small UAVs being constrained by onboard energy due to their limited size can be significantly enhanced by specific flight strategies according to expected atmospheric formations. Most of the energy harvesting methods rely on active control system that detect and exploit the energy of atmospheric turbulence through intentional maneuvering of the aircraft. As a response for such a request this paper propose one of the methods for wind estimation which could be used as a direct input of control for energy harvesting strategy. Moreover, such a system could be replicated on various positions along wingspan which could provide necessary information on gust length scales during the flight. The major benefit of such a biologically inspired sensory system is that the flying vehicle senses the disturbances rather than its responses to it.

An approach based on Unscented Kalman Filter (UKF) is proposed by Condomines [1] for non-linear wind estimation in aspect of formation detection of cumulus-type clouds with a fleet of drones. Review and suitability of conventional sensors applicable to small UAVs is performed by Mohamed [2]. The use of pressure sensors on the wing as a stabilization system of a micro UAV for roll axis was demonstrated by Mohamed [3, 4]. Another way of stabilizing a small UAV

*PhD Candidate, Department of Aerodynamics, Energetics and Propulsion, (nikola.gavrilovic@isae.fr)
‡Assistant Professor, Applied Aedrodynamics URI-Drones
‡Full Professor, Department of Aerodynamics, Energetics and Propulsion
⊥Associate Professor, Department of Conceptual Design of Aerospace Vehicles
△Associate Professor, Department of Conceptual Design of Aerospace Vehicles

in turbulent conditions has been performed by Mohamed [5] with pitch probes (multi-hole probes) located on both sides of the wing. Both ways promise more effective stabilization of the aircraft when compared to conventional inertial systems. Moreover, a correlation between a single pressure tap on the wing and cord-wise integrated pressure coefficient was used for roll mitigation of oncoming turbulence by Marino [6, 7]. Capacitive strip sensors applied on the airfoil skin was demonstrated by Callegari [8]. A stabilization systems based on surface pressure measurements can be considered as feeling way of turbulence affecting aircraft. They promise more effective response as opposed to conventional approach of traditional systems based on inertial sensors. The systems previously mentioned replicate the function of feathers and hairs as shown in Figure 1. A flush air data system intended for wind vector sensing in dynamic soaring UAVs is presented in [9]. The system uses pressure holes on the aircraft nose-cone as inspiration from Wandering Albatross and Giant Petrel nostrils. As opposite to fluffy wing structures, some birds are also equipped with rigid sensory systems as explained in previous work. An overall view on biologically inspired aerodynamic structures and their purpose is explained by Rasuo [10].

The advantage of the principle proposed in this paper is capability to estimate the precise value of angle of attack on the arbitrary chosen locations on the wing. Those precise information can be further used for meteorological investigation or as an direct input of control for energy extraction flight strategies of soaring or powered flight. The system is particularly interesting for soaring strategies as it allows aircraft to feel upcoming disturbances. It also provides insurance that detected turbulent structures can be exploitable due to their magnitude frequency and length scale.

2 Existing Meteorological Knowledge of Atmospheric Boundary Layer

Mini and micro UAVs predominantly fly in the surface of atmospheric boundary layer, where turbulence is strongly influenced by surface conditions, both terrain and temperature. We can differentiate two type of sources affecting turbulent formation of atmosphere. Mechanical is caused by friction as air flow masses move over the earth surface. Appearance of gradients in velocity will induce the formation of shear layers. Those shear layers produce rotating air motions or eddies and their strength is directly proportional to the magnitude of air velocity. The other sources could be the roughness and natural obstacles that deflect air flows. On the other hand thermal turbulence is caused by buoyancy effects. Unequal heating of the ground provoked by the clouds or natural obstacles as cliffs, mountains and valleys generate large circulation systems called thermals, where warmer air have a tendency of climbing while being replaced by cold air from the bottom. Those thermal irregularities are actually magnifying vertical mixing caused by mechanical turbulence. These two distinct sources present obvious challenge to flight stability and control but they also provide the opportunity and energy source for soaring flight strategies.

The measurements of turbulence in lower part of atmospheric boundary layer (region influenced by the frictional effect of surface extending from surface up to the range of 100 up to 1000 m depending on the surface and climate conditions) is usually done at fixed mast locations near wind turbine stations. It is known that turbulence intensity increases nearing the ground, strongly influenced by the terrain, thus changing the conditions comparable with those at high altitudes. Designation of standard deviation T_i of fluctuating velocity σ_u divided by mean velocity $\overline{U_z}$ for different altitudes and terrains shows that inhabited areas have the highest T_i of up to 50%. Although fluctuations are mainly present in the horizontal plane, while vertical components are mainly reduced but still present in the last couple of meters.

The average magnitude of wind in Europe measured in the horizontal plane at 10m above ground followed over the period of 44 years (1957-2002) show a variation of 2-4 $\mathrm{m{\cdot}s^{-1}}$ depending on the exact location of region as presented in [11]. Recent experiments from Watkins [12] considered measurements of spatial variation in pitch angle and confirmed statement about spatial variation of turbulence magnitude.

Previously described environment satisfies possible scenario of MAV flight concerning both altitude and experienced wind conditions. Moreover, it proves that atmospheric influence on low altitude UAV flight is three-dimensional.

3 Inspiration from Nature

Various styles of flight could be noticed while bird-watching. According to [13] birds use several strategies of energy harvesting, which serve as an inspiration for all the current improvements in the field of UAV long endurance performance. Interaction of wind and obstacles such as buildings, hills or waves generates an ascending component of air motion. Many birds with knowledge of soaring techniques use these up-drafts to power their flight instead of wing flapping. In case of unequal heating of earth's surface provoked by ex. punctured cloud layer, implies uplift of hot air, known as thermal. Eagles, condors, vultures and many other large birds use these up-drafts with a technique called thermal soaring in order to extend their endurance while searching for a prey. Another example is sweeping flight within the gust pushed by the waves. Gulls and pelicans use these gusts to power their flight by flying along the wave cliffs. Gaining speed while wave slows down, they are able to pull up and glide to another wave where the process continues. Some birds such as kestrels, remain motionless above a point on the ground by flying into the wind at a speed equal to that of the wind. This technique is called wind hovering.

There is a dense network of nerves around feather follicles according to [14]. Feathers are actually connected to the follicles in the skin and they represent a very complex system

of muscles and nerves interconnected. Primary function of such an anatomical configuration is mechanoreception. Specialized feathers on the head and breast have been shown to act as indicators of wind speed and direction. It has been also found that birds have very sensitive nerve endings (Herbst corpuscles) in their skin which are able to detect very high frequencies of vibrations of more than 100 Hz. Severe turbulent flows will cause the feathers to vibrate and gyrate wildly. As the feathers are elevated by the air stream, mechanoreceptors increase their discharge frequency according to [15].

Micro-structures require much more attention because they serve a lot of functions. So far understanding of flight control in birds is very limited. The connection between the natural sensory system and motor pathways involved in complex movements in soaring flight strategies have not yet been fully understood.

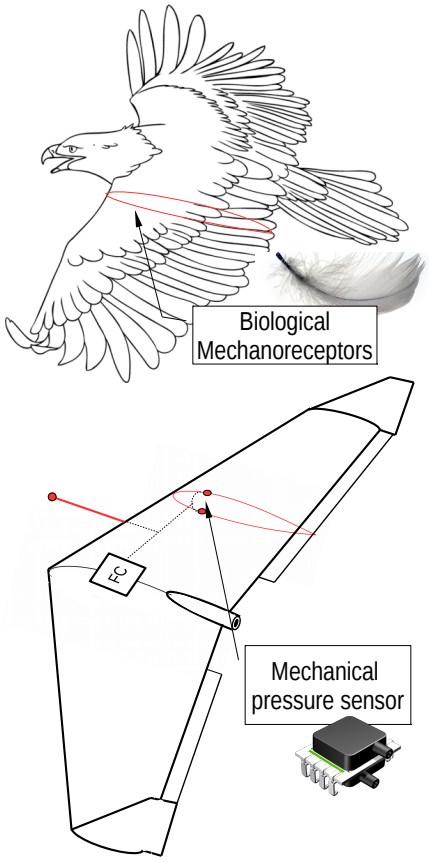


Figure 1: Bionical approach.

4 Algorithm Development

4.1 Wind Model for Simulations

From field experiments it is well known that undisturbed wind velocity is variable in space and time. The most adequate method to simulate a turbulent wind field would be to solve Navier-Stokes equations of an atmospheric flow bounded from below by an aerodynamically rough surface. This method requires enormous computational resources. Alternative could be Large Eddy simulations (LES) as an approximate solution to the Navier-Stokes equations where the smallest scales are not solved directly but modeled. Still even simplified alternative requires big computational power. Therefore, empirical description is generally used for turbulence representation, using spectral and coherence functions.

A model of turbulent wind field suitable for calculations requires good representation of both temporal and spatial structure of turbulence. Method for a generation of a single wind time series from a Kaimal spectrum is proposed by Branlard [16]. It leads to natural representation of turbulent flow of high computational cost compared to alternative Large Eddy or Navier-Stokes simulations. The spectrum used in simulations is presented by Kaimal [17] with its adjustable constants that depend on the chosen turbulent length scales, intensity of turbulence, surface roughness, Reynolds and Richardson number.

$$F_\alpha(k_1) = \frac{A \cdot \sigma_u^2 / k_{1m}}{1 + B \cdot (k_1/k_{1m})^{5/3}} \tag{1}$$

The characteristics of generated profiles are compared with available database on wind characteristics that can be found in [18] with intention to match the same level of turbulence energy (see Figure 2).

4.2 Numerical Calculations

After the generation of wind profile time series, two dimensional computational domain has been built with structured mesh around an airfoil SD2048. Structured mesh convergence has been studied previously to satisfy required number of cells for precise representation of boundary layer and wake formation. The chosen airfoil is a typical low-Reynolds number foil which could be found on several gliders including SB-XC used by NASA. The resulting vertical and horizontal wind profiles generated previously have been used as a direct input for inlet boundary condition of unsteady RANS simulations. The time step chosen for unsteady simulations was 0.5×10^{-4}s. The intention is to investigate the pressure variation on multiple locations of the airfoil and find a suitable way of achieving coherence with upcoming wind velocity or angle of attack. Moreover, the position of the pitot tube ahead of the airfoil was studied for various angles of attack.

The intention was to pick a specific pair of pressure ports where one port is on the upper surface of the airfoil and the other on the lower surface. Chosen pair has been selected with a request of precise estimation of angle of attack at high mean angles of attack, even after beginning of stall. Those points are recording a pressure with time, while the points in front of the airfoil record dynamic pressure (see Figure 3 for point location). The information on the pressure difference

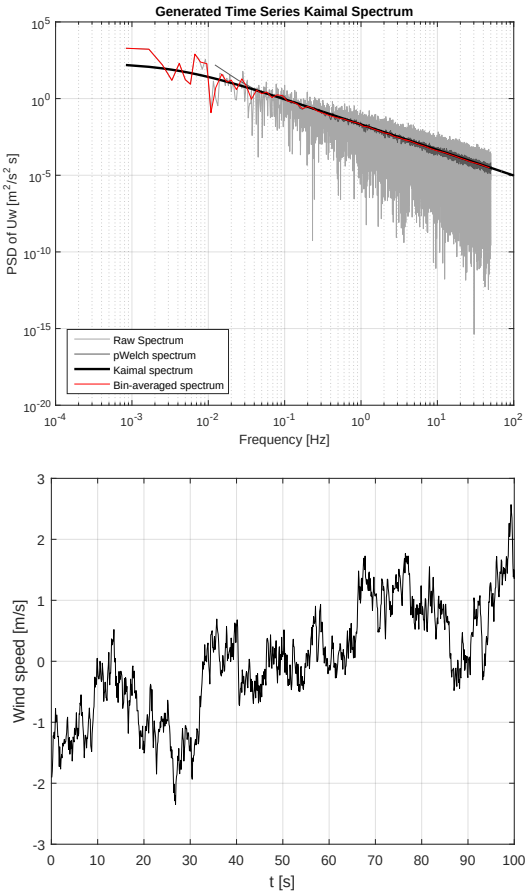


Figure 2: Generated time series of Kaimal spectrum as an input for URANS simulations.

is afterwards transformed into difference of pressure coefficients between the chosen ports. The following relation has been found:

$$\Delta C_{p12} = C_{p1} - C_{p2} = \frac{p_1 - p_2}{q} \tag{2}$$

The imposed wind profile at the inlet of the computational domain will generate additional angle of attack on the airfoil. The idea is to capture the angle of attack increments provoked by wind with related pressure coefficient fluctuations on the airfoil. This is achieved with polynomial fitting, where the optimization of the coefficients is performed with method of least squares.

$$\alpha = C_0 + C_1 \cdot \Delta C_{p12} + C_2 \cdot \Delta C_{p12}^2 \tag{3}$$

The relation claims that we are able to estimate the angle of attack knowing the pressure difference between the upper and lower point on the airfoil and dynamic pressure as shown in Figure 4. Once optimized for a certain airfoil and position of ports, coefficients of fitting were tested with various mean angles of attack and airspeeds. Those tests have shown that impact of Reynolds number variation does not affect significantly the angle of attack estimation.

However, for a variety of reasons it is clear that numerical simulations if posed correctly represent idealistic case where all the information of the flow are known in every node of the domain. Therefore, we proceed to realistic study of equipment in the flight test.

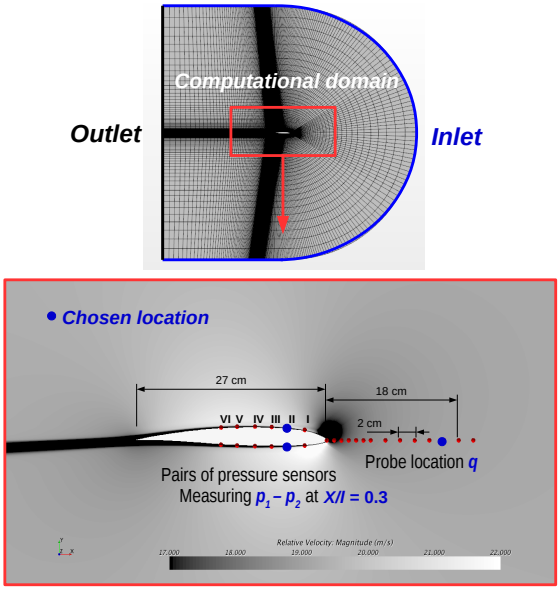


Figure 3: CFD computational domain of SD2048 Airfoil with port pair measuring pressure difference and pitot tube measuring dynamic pressure.

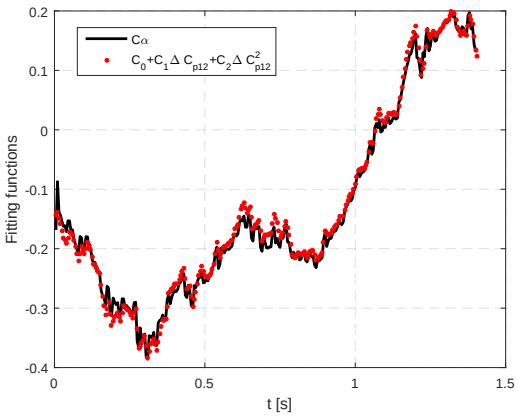


Figure 4: Curve fitting for numerical simulations ($Re = 340,000$).

5 Platform

5.1 UAV and Equipment

The chosen flight test vehicle was a flying wing shown in Figure 5 built in UAV laboratory of ENAC. Particular interest of using flying wing configuration is sufficient thickness of the wing cord for sensor and equipment integration.

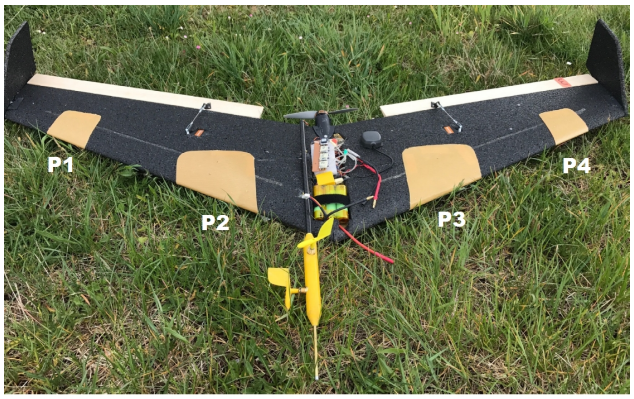


Figure 5: Flying wing UAV with equipment for angle of attack estimation.

Description	Details
Airfoil	Sipkill 1.7/10
Wingspan	1.2 m
Mean Aerodynamic cord	0.27 m
Weight	0.8 kg
Cruise speed	12 m/s
Aspect ratio	7

Table 1: Aircraft characteristics.

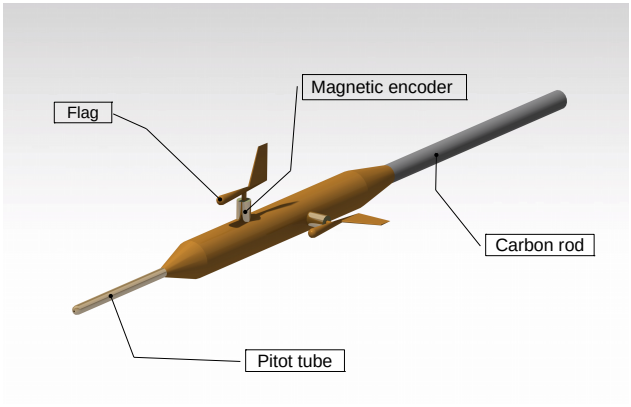


Figure 6: Pitot tube and wind vane with magnetic encoder.

Pressure sensors are located beneath the kevlar wing skin at 30% of cord distance. Small holes of 2mm diameter were made in vertical plane of the wing. Totally four places were chosen marked as points 1,2,3 and 4 on Figure 5. The distance between points is 40 cm. Each port pair is connected to a single differential pressure sensor.

Carbon rod has been joined to a 3d printed housing on its tip. The size of the housing was designed in a way to accept magnetic encoder, pitot tube and all necessary wiring and pressure tubes (see Figure 6). The length of the carbon rod was previously determined in CFD simulations for a non-disturbed pressure field condition. Small, 3d printed flag was attached to magnetic encoder. All of the sensors shown in Table 2 work at high frequency of 50Hz and recorded to on-board data logger except the differential GPS which works at 5 Hz.

Description	Details
Autopilot	Paparazzi [19] Chimera v1.0
IMU	MPU-9150 based
DGPS	NEO-M8P2
Differential pressure sensor	HCLA02×5EB
Magnetic Encoder	MA3-P12-125-B
Wind vane	3D printed
Pitot tube	10 cm

Table 2: Equipment on-board.

5.2 Calibration of Sensors

Calibration of the pitot tube, pressure sensors and the magnetic encoder has been performed in a wind tunnel with an autopilot on-board. Reference for pitot tube was imposed velocity of the wind tunnel measured with a hot wire anemometer. Calibration of pressure and flag sensor was done with respect to the IMU output from the autopilot, due to the equality of climb angle and angle of incidence in the wind tunnel. Curve fitting has been performed once again with least square method in PYTHON and obtained constant coefficients were used for further flight tests analysis. One of the drawbacks of this system is that pressure sensor (for angle of attack estimation) calibration is strictly related to the wing geometry of the pressure port location. Once calibrated, sensor for chosen aircraft and position is not reusable for a different wing shape and dimensions. Due to the required precision, ground and climb speed estimation could not rely on only GPS output. Especially problematic estimation of altitude requires combined work of barometer, differential GPS and accelerometer.

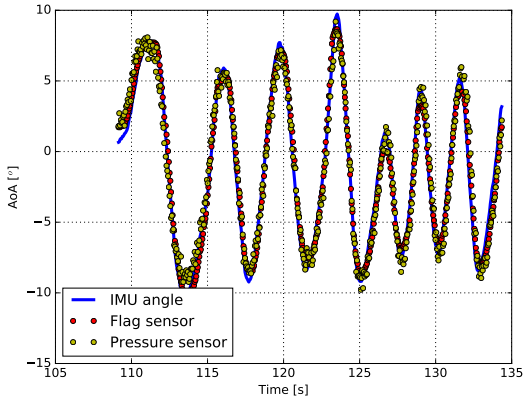


Figure 7: Results of calibration from wind tunnel experiment

6 Analysis of Experimental Data

6.1 Flight Data

Our intention is to use available information of flight parameters for further processing and transformation into wind components. The process of wind estimation requires knowledge of climb angle θ coming from IMU system, angle of attack α coming from pressure or flag sensors, dynamic pressure q coming from pitot tube and finally ground and climb speed coming from GPS, barometer and accelerometer combined together. With respect to the Figure 8 we write following equations for wind field estimation:

$$w_x = \dot{x}_i - V\cos\gamma \tag{4}$$

$$w_z = \dot{z}_i + V\sin\gamma \tag{5}$$

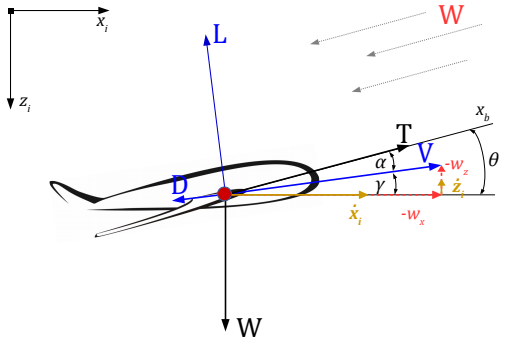


Figure 8: Longitudinal equations of motion.

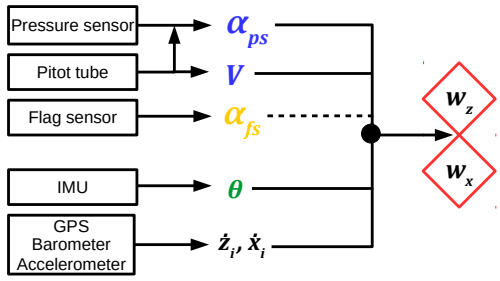


Figure 9: Algorithm for wind components estimation.

The flight has been started directly in autopilot assisted mode. After a short attempt, a small angle of attack increment was required in order to sustain the level flight. Due to the fact that ailerons occupy most of the wingspan, certain modifications in algorithm function have been made taking into account aileron deflection.

$$\alpha = C_0 + C_1 \cdot \Delta C_{p12} + C_2 \cdot \Delta C_{p12}^2 + C_3 \cdot \delta_a \tag{6}$$

Some intentionally provoked oscillations were made in order to visualize similarity between the angle of attack estimated by pressure and flag sensors. The following Figure 10 shows acceptable similarity between the two with a slight difference in magnitude. The difference is coming due to the fact that 3D printed flag has a certain inertia and due to rapid changes in angle of attack it shows a small increment.

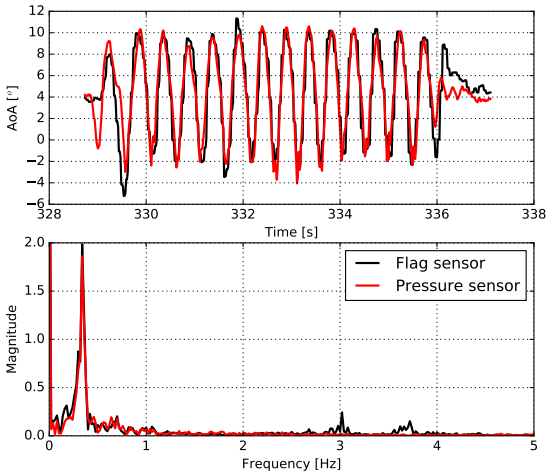


Figure 10: High frequency oscillations in flight test provoked by command input in calm atmosphere

Most of the time, flight test resulted with negligible error between angles estimated by different point locations (See

Figure 11). This is due to the fact that flying was conducted in relatively calm atmosphere with airspeed of 12 m/s. However, there were some parts where error was considerable. One of them is shown in Figure 12 where relative difference computed with respect to point 1 located at far right side of the wing is highest for point 4 located at opposite side of the wing. Described discontinuity results in roll and yaw moment regulated by actuators for autonomous flight regime. The discontinuity is coming from the small length scale gust, thus implying different wind components seen by each side of the wing locally. It is clear from Figure 12 that left side of the wing (Points 3 and 4) saw higher vertical wind component as this part of the wing registered higher local angle of attack. Locally, angle of attack were dropping till Point 1 with lowest amplitude.

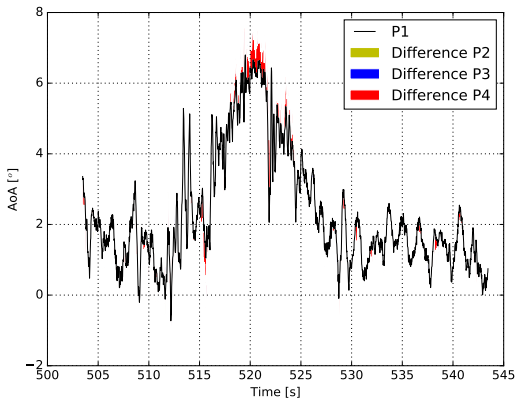


Figure 11: Local AoA difference compared to first point

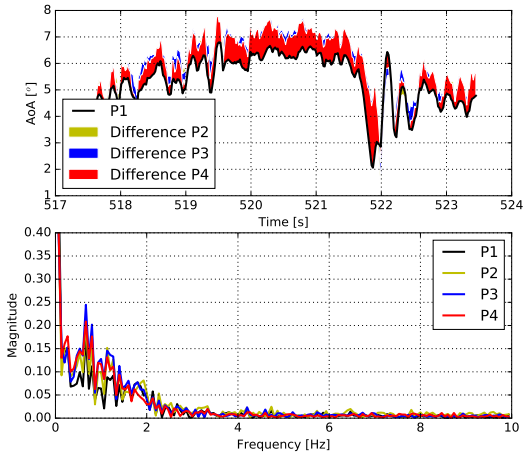
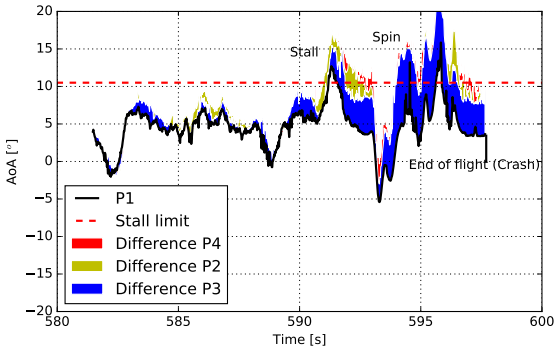


Figure 12: Local AoA difference compared to first point - zone of low coherence

Input for the control stabilization in this case is regulated from the IMU coming from autopilot. On one side, IMU acts as a correcting system which responds to direct consequence of disturbance, while on the other local angle of attack estimation promise "feeling" of upcoming disturbance pointing towards more effective way of control.

Advantage of having multiple location angle of attack sensors can also be found in stall control and evasion which was demonstrated by Bunge [20].

The following flight case can be divided into three phases. First phase represents partial stall of the right side of the wing as points 1 and 2 first reached stall limit. This led to an immediate, unrecoverable spin of the aircraft. Despite the efforts of autopilot to recover the plane, as he was leaving stall several short periods, right part of the wing was still reaching stall limit which resulted in a crash. The potential of these information could be easily implemented into the autopilot control laws, restricting the exceeding of stall limit on any part of the wing.



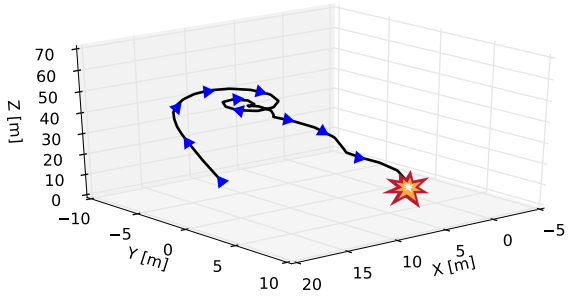


Figure 13: Stall, spin and crash

Moreover, the case of wind field estimation used as a control input for gust energy-extracting strategy can be a decisive mechanism for control activation as energy retrieval presented by Gavrilovic [21] is guaranteed in case of length scales greater than plane. Accurate estimation of wind field components depends on precision of all the elements involved as discussed in the beginning of this section. So far the estimation relies on available equipment where the weakest link

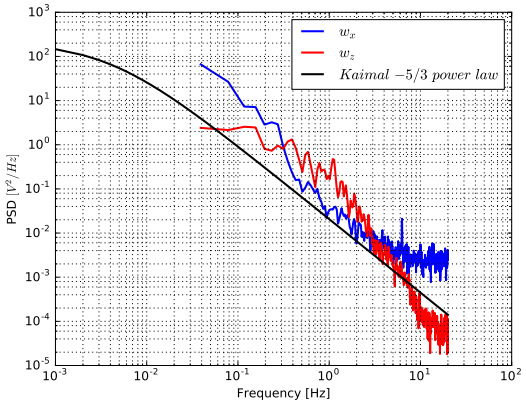


Figure 14: Estimated wind spectrum

is certainly the climb speed estimated by GPS. As shown in Figure 14 both vertical and horizontal component follow the natural law of turbulence dissipation represented by Kaimal spectrum [17] in this case. However, convincing statement on accuracy of wind components estimation requires comparison between estimated wind field by plane and available data from another source (example of flying around meteorological mast at different altitudes) which will be the subject of our further study.

7 Conclusion

Unlike the majority of the wind field data sets, the tests presented here are related to a typical mini UAV flying environment in the low-level of atmospheric boundary layer. The presented work shows development of a system for angle of attack estimation based on pressure measurements on the wing for further investigation of meteorological conditions experienced by a small UAV. The system showed several potential applications. The current results have concentrated on the spatial variation in angle of attack along an aircraft wing span. Ability to locally estimate the angle of attack promise potential for control of upcoming roll motions of the plane before the inertial response, as information on local angle of attack can be used as direct input of active control for stabilization. The difference between the local angles of attack have been clearly identified during the flight tests. Moreover, the system can be involved in a stall prevention mechanism of the autopilot. As critical point of stall on the wing depends on its geometry, those locations were used for pressure ports, thus detecting initial separations. Particular interest will be the implementation of algorithm for wind field estimation. Beside the knowledge of wind field components, the system will also provide a decisive mechanism for actions for power gain in energy harvesting flight strategies. Results have also shown acceptable comparison of measured and theoretical wind spectra.

Acknowledgements

Part of the presented UAS operations have been conducted at the Lannemezan experimental site belonging to the Pyrenean Platform of Observation of the Atmosphere (P2OA, hhtp://p2oa.aero.obs-mip.fr), a research platform of the University Paul Sabatier, Toulouse (France).

References

[1] Condomines, J-P., Bronz, M., Hattenberger, G. and Erdelyi, J-F., " Experimental Wind Field Estimation and Aircraft Identification," IMAV 2015, 2015.

[2] Mohamed, A., Clothier, R., Watkins, S. and Abdulrahim, M., "Fixed-wing MAV Attitude Stability in Atmospheric Turbulence, Part 1: Suitability of Conventional Sensors", *Progress in Aerospace Sciences*, 70, 2014.

[3] Mohamed, A., Watkins, S., Clothier, R. and Abdulrahim, M., "Influence of Turbulence on MAV Roll Perturbations," *International Journal of Micro Air Vehicles*, Vol 6, Number 3, 2014.

[4] Mohamed, A., Watkins, S., Fischer, A., Marino, M., Massey, K. and Clothier, R., "Bioinspired Wing-Surface Pressure Sensing for Attitude Control of Micro Air Vehicles," *Journal of Aircraft*, 2015.

[5] Mohamed, A., Abdulrahim, M., Watkins, S. and Clothier, R., "Development and Flight Testing of a Turbulence Mitigation System for Micro Air Vehicles," *Journal of Field Robotics*, 1-22, 2015.

[6] Marino, M., Watkins, S., Sabatini, R. and Gardi, A., "Unsteady Pressure Measurements on a MAV Wing for the Design of a Turbulence Mitigation System," IEEE, 2014.

[7] Marino M., "Unsteady Pressure Sensing on a MAV Wing for Control Inputs in Turbulence," PhD thesis, RMIT University, 2013.

[8] Callegari, S., Tallameli, A., Zagnoni, M., Golfarelli, A., Rossi, V., Tartagni, M. and Sangiorgi, E., "Aircraft Angle of Attack and Air Speed Detection by Redundant Strip Pressure Sensors", Sensors, IEEE, 2004.

[9] Quindlen, F. and Langelaan, J., "Flush Air Data Sensing for Soaring-Capable UAVs," AIAA 51st Aerospace science meeting, Grapevine, TX, USA, 2013.

[10] Rasuo, B., "Flight Mechanics," Faculty of Mechanical Engineering, University of Belgrade, Belgrade, (in Serbian) e-book, 2014.

[11] Peter, K., "Analysis of the European Wind Power Climatology and the Possible Cosmic Radiation Forcing on Global Lightning Activity," PhD Thesis, Budapest, Hungary, 2009.

[12] Watkins, S., Milbank, J., Loxton, B. and Melbourne, W., "Atmospheric WInds and Their Implications for Microair Vehicles," *AIAA Journal*, Vol. 44, No 11, 2006.

[13] Scott, S. and McFarland C., "Bird Feathers - A Guide to North American Species," Stackpole Books, 2010.

[14] Videler, J., "Avian Flight," Oxford Ornithology Series, Oxford University Press, New York, 2005.

[15] Brown, R. and Fedde R., "Airflow Sensors in the Avian Wing," *Journal of Experimental Biology*, 179, 13-30, 1993.

[16] Branlard, E., "Generation of Time Series From a Spectrum: Generation of Wind Time Series From the Kaimal Spectrum, Generation of Wave Time Series From Jonswap Spectrum," Technical University of Denmark, 2010.

[17] Kaimal, C. and Finnigan, J., "Atmospheric Boundary Layer Flows, Their Structure and Measurement", Oxford University Press, New York, 1994.

[18] Database on wind characteristics (online database), Dept. of Wind Energy, Technical Univ. of Denmark, Lyngby, Denmark, *winddata.com*.

[19] Hattenberger G., Bronz M. and Gorraz M., "Using the Paparazzi UAV System for Scientific Research," IMAV 2014, International Micro Air Vehicle Conference and Competition, 2014.

[20] Bunge, R., Alkuardi, A., Alfaris, E. and Kroo I., " In-Flight Measurement of Wing Surface Pressure on a Small-Scale UAV During Stall/Spin Maneuvers," AIAA Flight Testing Conference, Washington D.C., USA, 2016.

[21] Gavrilovic, N., Benard, E., Pastor, P. and Moschetta, J-M., "Performance Improvement of Small UAVs Through Energy Harvesting Within Atmospheric Gusts," Atmospheric Flight Mechanics Conference , Grapevine, TX, USA, 2017.

Developing a Stable Small UAS for Operation in Turbulent Urban Environments

R. Gigacz*, A. Mohamed*, P. Poksawat, S. Watkins and A. Panta
RMIT University, Melbourne

ABSTRACT

The stability of Small Unmanned Air Systems (SUASs) can be challenged by turbulence during low-altitude flight in cluttered urban environments. This paper explores the benefits of a tandem wing aircraft configuration with the implementation of a pressure based phase-advanced turbulence sensory system on a SUAS for gust mitigation. The objective was to utilize passive and active methods to minimise gust-induced perturbations. Experimentation in repeatable turbulence within a wind tunnel's test section was conducted. The experiments focus on the roll axis, which is isolated through a specially designed roll-axis rig. The results shows improvement over conventional aircraft. This work is part of a larger research project aimed at enabling safe, stable and steady SUAS flight in urban environments.

1 INTRODUCTION

Small Unmanned Air Systems (SUASs), or Micro Air Vehicles (MAVs) typically operate at low altitude, within the atmospheric boundary layer. This region is optimum for a range of SUAS applications in Intelligence, Surveillance, and Reconnaissance (ISR) missions and is characterised as having high turbulence intensity [1, 2]. In the presence of winds, SUAS performance can degrade significantly [3]. However, turbulence poses an even greater threat to the vehicle's attitude stability [4–7]. Consequently, attitude control in turbulence is a critical issue for SUASs and MAVs as identified by Mohamed, Massey, Watkins and Clothier [8]. Current attitude control systems can be challenged by atmospheric turbulence [9]. Sensors which have the ability to detect the oncoming gusts could potentially complement or replace conventional inertial-based sensors for robust attitude control [10, 11]. Phase-advanced multi-hole pressure based sensors which are able to react to the turbulence ahead of the leading edge have been developed and patented [12], these have been shown to increase the stability of SUASs in turbulence [13].

This paper explores passive and active methods of aiding the stability of SUASs through experimental wind tunnel testing of a tandem wing airframe, in conjunction with the phase-advanced multi-hole pressure probes, to further enhance the attitude control performance in high levels of turbulence. Roll perturbations were identified as the most significant disturbing factor for small fixed-wing aircraft [1, 14] and consequently will be the focus of this paper.

*Email address(es): Rohan.Gigacz@gmail.com, Abdulghani.Mohamed@rmit.edu.au

2 TANDEM WINGS

The concept of a tandem wing aircraft is not a new one, Minardo and Trainelli [15] compiled a report with many examples of tandem wing aircraft which have been produced, noting how the first Wright Flyer in itself was partially a tandem wing, being a Canard configuration aircraft. A tandem wing aircraft is described as being an aircraft with two independent lift generating wings, which eliminates the need for a conventional horizontal tail and elevator. In order for it to remain a true tandem wing aircraft and not a canard aircraft, both wings must be of similar wingspans, and they will generally be set on different planes separated vertically and/or horizontally. It should be noted although many different civilian tandem wing aircraft have been built over time, the design has not become popular. In recent times the tandem wing configuration has started to make a resurgence, unmanned aircraft can potentially benefit from a tandem wing configuration and it has been implemented successfully in various aircraft (e.g., ADCOM Systems Yabhon, Aeronvironments Switch Blade and Innocons MicroFalcon).

There have been various studies on Low Reynolds number tandem wing airfoils, such as [16] along with [17] which primarily looked at optimization of a tandem wing design, but not in relation to flight through high levels of turbulence. The tandem wing configuration has a number of hypothesized advantages with respect to counteracting turbulence and when used in conjunction with the phase-advanced multi-hole pressure sensor system, as outlined in Figure 1. They are are well suited for precision active control through the control surfaces embedded in its wings providing higher control authority and an added degree of freedom (heave). Heave is a translational vertical movement up or down, which is created by activating both the control surfaces on the front and rear wings to an extent which would provide an equal lift on both sets of wings, thus creating a heave motion. Heave has been identified as a common perturbation for SUASs and MAVs when in turbulence with length scales that are greater or equal to its wing span [8,18]. The tandem wing design would appear to counter heave perturbations without the need to change the aircrafts pitch angle as much as is the case with conventional configu-

rations. This can be particularly useful for on-board payloads which can be blurred due to rotational motion. Furthermore, the implementation of advanced pressure sensors [10, 11, 13] on tandem wing designs has the potential to enhance the time advantage by allowing the placement of the roll control surfaces (ailerons) further aft of the probe sensors and enables a number of control architectures to be utilised.

3 Vehicular Development

A tandem wing aircraft has been developed (Figure 2 and 7) which utilises the phase-advanced multi-hole pressure sensors ahead of the front wings leading edge, as used in [13]. The specifications of the aircraft are outlined in Table 1 and 2. Due to the high frequency demands of stable flight in a high level of turbulence, high performance servos must be utilised for movement of the control surfaces. An all metal gear servo was selected, the metal gears are necessary to allow for sustained high frequency and load movement of the servos without overheating. Specifications of this servo are outlined in Table 3.

A fixed span, flat-plate airfoil was selected for the experimental aircraft, it's performance in smooth flow has has been documented by Mueller [19].

The sizing of the aircraft and its wings were made to be representative of typical SUASs which can be handled and launched by one person.

For the purpose of comparison, a fixed span flat-plate horizontal stabilizer of 40 % of the main wing was used. This additional horizontal stabiliser replaced the rear wing, to represent a similarly sized conventional aircraft of the same wing span. This conventionally winged alteration of the tandem wing SUAS is shown in Figure 3.

Characteristic	Detail
Airfoil	Flat plate
Leading Edge	Ellipsoid
Wing length	290.0 mm
Wing-span	650.0mm
Chord	150.0 mm
Camber	4.0 mm
Cruise speed	9 m/s
Wing spacing (LE - LE)	450 mm

Table 1: Aircraft specifications.

3.1 Control System

A custom developed flight controller was used to control the aircraft's attitude. The main components are a 32-bit ARM processor and an inertial-measurement-unit (IMU), which are required for the attitude estimation and real-time control implementation. A nonlinear complementary filter, presented in [20], is implemented to compute the aircraft's attitude. In addition to the conventional IMU based control architecture, the differential phase-advanced pressure based system as previously described is implemented to react to the turbulence ahead of the aircraft. This system has been outlined in [13] and is implemented by placing the differential pressure probes ahead of the leading edge of each of the front wings. The measured pressure is used as a feed-forward into the inner loop control system of the cascaded PID controller, as shown in Figure 4, where Kff is the feed forward gain to be tuned experimentally.

Component	Detail
Processor	Teensy 3.2
IMU	MPU6050
Servo	RJX FS0435HV
Receiver	FrSky XSR 2.4 Ghz ACCST
ESC	Turnigy 30 A, SBEC 4 A 5 V
Battery	Turnigy 1200 mah 25-50 C 3 S
Pressure Sensor	Honeywell HSCDRRN005NDAA3
Data Logger	"Blackbox" Data Recorder
Voltage Regulator	DC-DC Stepdown Module

Table 2: Aircraft component specifications.

Characteristic	Detail
Operating Voltage	4.8 - 7.4 V
Torque	3.4 kg/cm @ 7.4 V
Speed	0.04 sec/60° @ 7.4 V
Frequency	333 hz
Gear Type	All Metal Gear
Weight	20 g

Table 3: RJX FS0435HV servo specifications.

4 Experimental Setup

Passive turbulence generation, using planar grids, represented the most suitable method for producing elevated levels of turbulence intensity inside a wind tunnel. RMITs Industrial Wind-Tunnel (2x3x9 m test section) was considered sufficiently large to simulate the relevant turbulence conditions of varying length scales and intensities, the aircraft in the tunnel with the turbulence grid can be seen in Figure 2 and 7. The approach outlined by Watkins, et al [21] is followed to characterize SUAS's and MAV's flight environment and replicate atmospheric turbulence. A Reynolds Number of $\approx$ 60,000 was tested representing typical SUAS flight regime. The selected turbulence intensities were 12.6 % and 18.0 % with a length scale of 0.31 m. The wind tunnel was operated at a speed which corresponds to 9 m/s at the aircraft's position, this airspeed value representative of typical SUAS and MAV flight speeds.

The primary focus of this paper is comparing the roll stability characteristics of the tandem wing aircraft with the

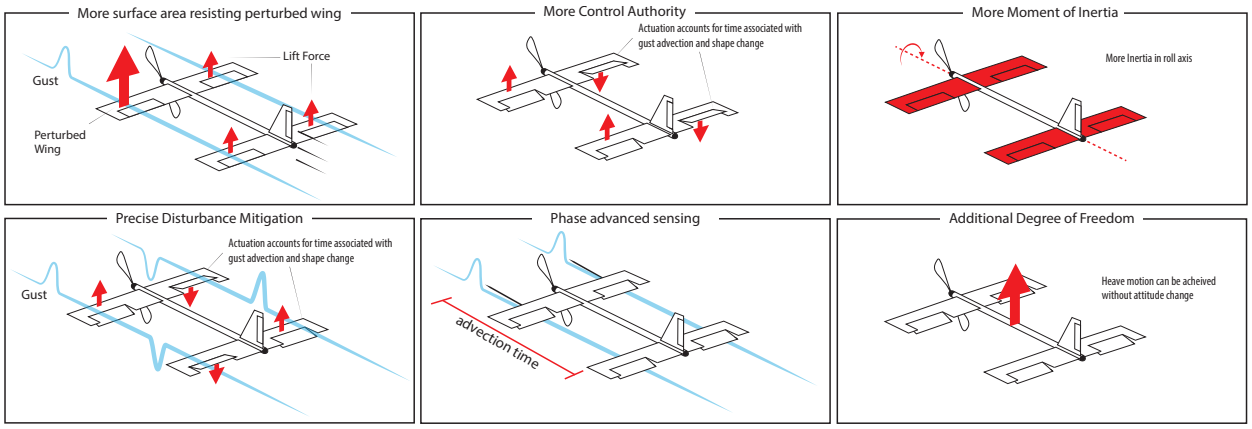


Figure 1: Tandem wing aircraft hypothesized advantages with respect to counteracting turbulence.

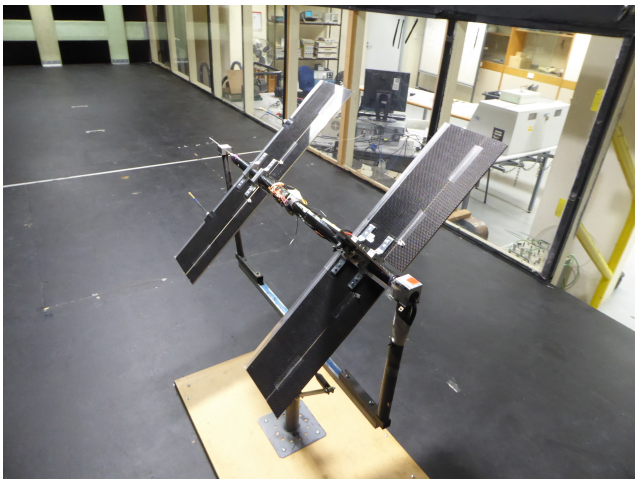


Figure 2: Tandem wing SUAS in roll rig.

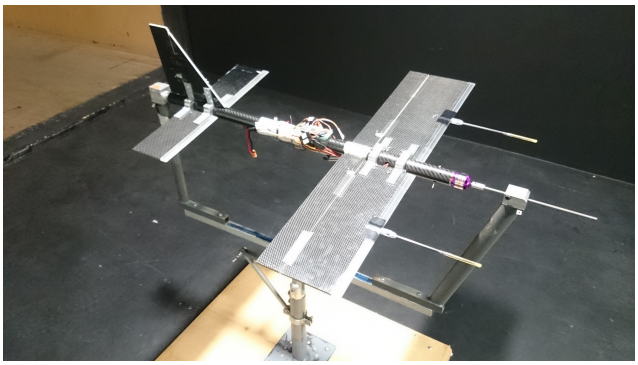


Figure 3: Conventional SUAS in roll rig.

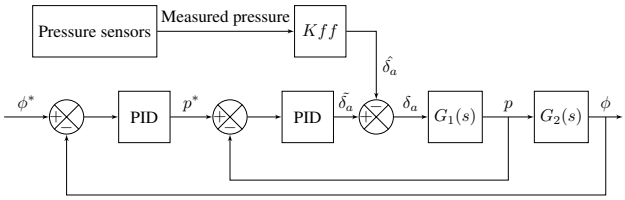


Figure 4: Pressure-based cascaded PID control structure for the roll axis.

pressure probes either active or inactive, with both the front and rear wing control surfaces acting as ailerons. Although for real flight of a tandem wing aircraft, the front wing may generally have a higher loading with the CG towards the front wing and thus some trim to the control surfaces would be needed, this study looks at the system with all control surfaces trimmed at a $0°$angle.

The aircraft's roll performance has been assessed through the aid of the roll axis rig detailed in [22]. The rig constrains the motion of the SUAS to a single axis, that of roll, with low friction.

4.1 Aircraft Control Loop Tuning

The roll axis PID gains were tuned in the selected experimental turbulence level of 12.6 % via a process of trial and error, where the final selection of each gains value was selected by running the aircraft for 60 seconds over a range of estimated gain values, and analysing the resulting data, with the value selected which corresponded to the least roll angle and roll rate perturbations. As this is a SUAS operating in high frequency turbulence the Derivative component of the PID controller is neglected, this is because the derivative term amplifies noisy signals [23]. This can also be better for the servos, as it may reduce the demand placed on them by reducing the frequency of actuation commands. Only the Proportional and Integral components of the PID control loop are needed.

Probability density functions (PDF) and boxplots were used to analyse the aircraft's performance while tuning in terms of roll angle and roll rate. The boxplot is a typical box and whiskers plot with a line for the median, '+' for the mean, a box around the 25 % and 75 % quartiles and whiskers bounding 2.5 % and 97.5 %. The PDF plots can be interpreted such that a lower distribution and higher peaks corresponds to a reduction in perturbations of roll angles and roll rates.

A similar process of trial and error was followed when tuning the pressure sensor control loop gains, Kff. An example of this tuning is shown through PDFs and boxplots in Figure 5 and 6, whereby the Kff value of 25 was initially selected as the most appropriate value as it corresponded to the least roll angle and roll rate perturbations. After this initial range was tested, the range of values were further lowered until a more precise value was obtained.

A similar process was followed for the conventional aircraft, however only the rate mode PID gains required change, with the attitude mode and probe gains able to remain the same.

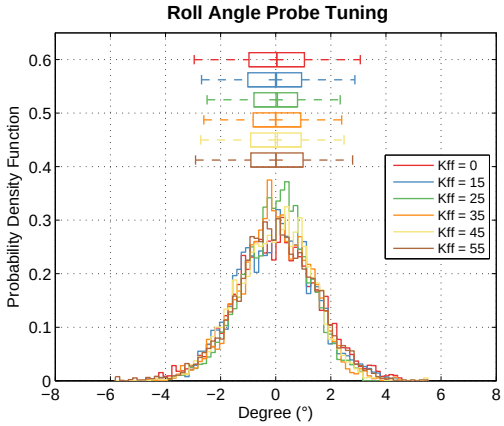


Figure 5: Roll angle probe tuning.

5 Results

PDFs and boxplots were used to analyse the aircraft's stability performance, where if the aircraft remained completely unperturbed in roll, the PDF plot would all be at 0°. Figure 8 and 9 show a comparison of the perturbations of the tandem wing and conventional aircraft in 12.6 % turbulence intensity, with and without the pressure probe system activated in the control loop. It can be seen through the smaller box plots, higher peaks and lower distribution that there is a reduction in perturbations when the pressure probes are activated for both the tandem wing and conventional aircraft. Through similar analysis of the PDFs and boxplots, it can be observed that the tandem wing aircraft has lower perturbations in the turbulence compared to the conventional aircraft.

In an effort to further explore the tandem wing aircraft's stability performance in high levels of turbulence, testing was

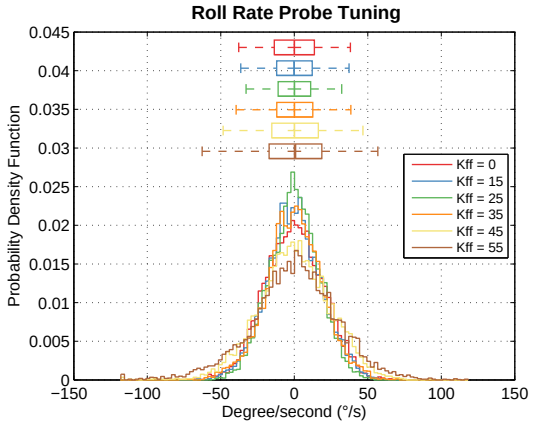


Figure 6: Roll rate probe tuning.

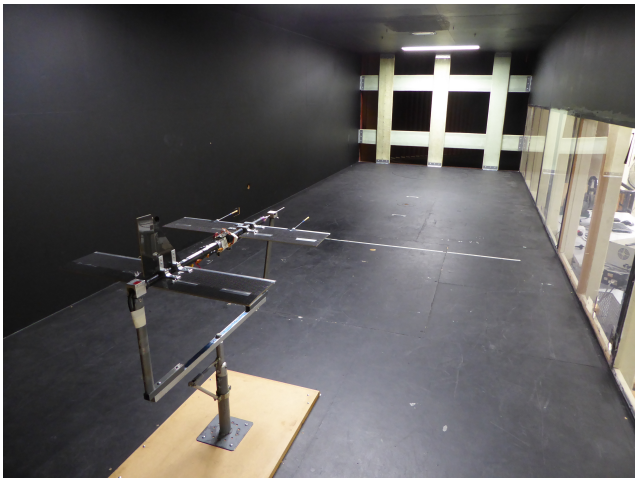


Figure 7: Tandem wing SUAS in roll rig.

conducted in 18 % turbulence intensity. PDFs and box plots of the aircraft's performance at this intensity are shown in Figure 10 and 11. Much like in 12.6 % turbulence intensity, there is a reduction in perturbations when the pressure probes were activated.

6 Conclusion

A tandem wing SUAS has been developed that is equipped with phase-advanced multi-hole pressure sensors to explore means of improving the attitude control in high levels of turbulence. Baseline performance and performance improvements have been demonstrated, emphasising the suitability of the tandem wing configuration in aiding safe and stable SUAS flight in turbulent urban environments. Much like past studies with conventional aircraft, utilising the phase-advanced pressure probes in conjunction with a standard PID control structure improves the roll stability of tandem wing SUASs in turbulence. Furthermore it has been

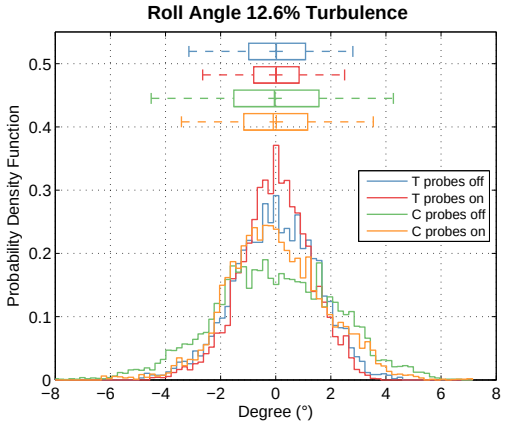


Figure 8: Roll angle perturbation of tandem wing (T) & conventional (C) aircraft with pressure probes off and on in 12.6 % turbulence intensity.

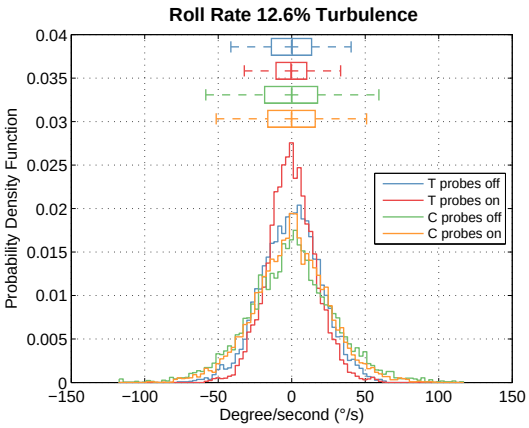


Figure 9: Roll rate perturbation of tandem wing (T) & conventional (C) aircraft with pressure probes off and on in 12.6 % turbulence intensity.

demonstrated that a tandem wing aircraft has lower roll angle and roll rate perturbations than a conventional aircraft with the same wing span in 12.6 % turbulence intensity. Future work will explore varying control architectures and configurations, comparisons with different conventional aircraft set ups, along with testing without the aid of a roll rig, including testing of the tandem wing configuration heave and pitch characteristics in turbulence.

Acknowledgements

The authors wish to acknowledge the technical support provided by Gilbert Atkins at RMIT University for setting up the experimental instruments in order to conduct the presented research. This research was undertaken as part of the RMIT Unmanned Aircraft Systems Research Team, within the Sir Lawrence Wackett Aerospace Research Centre, at RMIT University. This work has been partially funded and supported in part by; the U.S. Air Force Office for Scientific Research (AFOSR); Defence Science and Technology Group (DSTG); Defence Science Institute (DSI); and the Australian Government Research Training Program Scholarship.

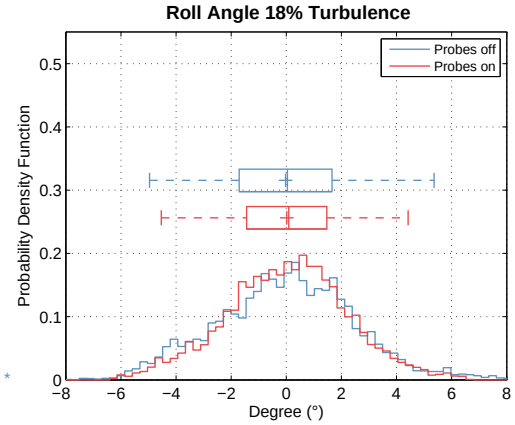


Figure 10: Roll angle perturbation of tandem wing aircraft with pressure probes off and on in 18 % turbulence intensity.

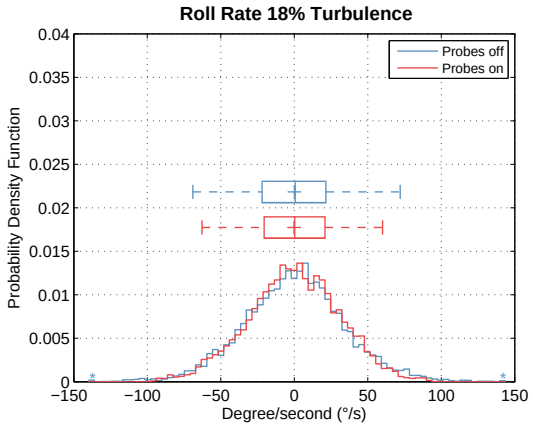


Figure 11: Roll rate perturbation of tandem wing aircraft with pressure probes off and on in 18 % turbulence intensity.

References

[1] S. Watkins, J. Milbank, and B. J. Loxton, "Atmospheric Winds and their Implications on Micro Air Vehicles," *AIAA journal*, no. September, 2006.

[2] S. Watkins, M. Thompson, B. Loxton, and M. Abdulrahim, "On Low Altitude Flight Through The Atmospheric Boundary Layer," *International Journal of Micro Air Vehicles*, vol. 2, no. 2, pp. 55–68, 2010.

[3] C. White, S. Watkins, E. W. Lim, and K. Massey, "The Soaring Potential of a Micro Air Vehicle in an Urban

Environment," *International Journal of Micro Air Vehicles*, vol. 4, no. 1, pp. 1–14, 2012.

[4] C. R. Galinski, "Gust Resistant Fixed Wing Micro Air Vehicle," *Journal of Aircraft*, vol. 43, no. 5, pp. 1586–1588, 2006.

[5] C. Galinski and R. Zbikowski, "Some problems of micro air vehicles development," *Bulletin of the Polish Academy of Sciences-Technical Sciences*, vol. 55, no. 1, pp. 91–98, 2007.

[6] P. Lissaman, "Effects of Turbulance on Bank Upsets of Small Flight Vehicles," *AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, no. January, pp. 1–10, 2009.

[7] W. Shyy, H. Aono, S. K. Chimakurthi, P. Trizila, C.-K. Kang, C. E. Cesnik, and H. Liu, "Recent progress in flapping wing aerodynamics and aeroelasticity," *Progress in Aerospace Sciences*, vol. 46, no. 7, pp. 284–327, 2010.

[8] A. Mohamed, K. Massey, S. Watkins, and R. Clothier, "The attitude control of fixed-wing MAVS in turbulent environments," *Progress in Aerospace Sciences*, 2014.

[9] A. Mohamed, S. Watkins, R. Clothier, M. Abdulrahim, K. Massey, and R. Sabatini, "Fixed-wing MAV attitude stability in atmospheric turbulence-Part 1: Suitability of Conventional Sensors," *Progress in Aerospace Sciences*, 2014.

[10] S. Watkins, A. Mohamed, A. Fisher, R. Clothier, R. Carrese, and D. Fletcher, "Towards autonomous MAV soaring in cities: CFD simulation, EFD measurement and flight trials," *International Journal of Micro Air Vehicles*, vol. 7, no. 4, pp. 441–448, 2015.

[11] A. Mohamed, S. Watkins, R. Clothier, M. Abdulrahim, K. Massey, and R. Sabatini, "Fixed-wing MAV attitude stability in atmospheric turbulence-Part 2: Investigating biologically-inspired sensors," *Progress in Aerospace Sciences*, 2014.

[12] A. Mohamed, S. Watkins, and R. CLOTHIER, "Methods and systems for attenuating the effects of turbulence on aircraft," Dec. 3 2015, wO Patent App. PCT/AU2015/000,326. [Online]. Available: https://www.google.com/patents/WO2015179905A1?cl=en

[13] A. Mohamed, M. Abdulrahim, S. Watkins, and R. Clothier, "Development and Flight Testing of a Turbulence Mitigation System for MAVs," *Journal of Field Robotics*, pp. 1–22, 2015.

[14] M. Abdulrahim, S. Watkins, M. Thompson, R. Segal, M. Shortis, and J. Sheridan, "Measurements of Gust Sensitivity on MAVs: Part2," *AIAA Guidance, Navigation and Control*, 2009.

[15] A. Minardo, "The Tandem Wing: Theory, Experiments, and Practical Realisations," Ph.D. dissertation, 2014.

[16] D. F. Scharpf and T. J. Mueller, "Experimental study of a low Reynolds number tandem airfoil configuration," *Journal of aircraft*, vol. 29, no. 2, pp. 1–6, 1992. [Online]. Available: http://arc.aiaa.org/doi/pdf/10.2514/3.46149

[17] Z. Husain, M. Abdullah, and T. Yap, "Two-dimensional analysis of tandem/staggered airfoils using computational fluid dynamics," ... *Journal of Mechanical* ..., vol. 33, no. 3, 2005. [Online]. Available: http://manchester.metapress.com/index/X10L92T311420555.pdf

[18] R. C. Michelson., "Overview of micro air vehicle system design and integration issues," *Encyclopedia of Aerospace Engineering*, 2010.

[19] T. Mueller, "Aerodynamic measurements at low reynolds number for fixed wing micro-air vehicles." 2000.

[20] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, "A complementary filter for attitude estimation of a fixed-wing UAV," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 340–345, 2008.

[21] S. Watkins, A. Fisher, A. Mohamed, M. Marino, M. Thompson, R. Clothier, and S. Ravi, "The Turbulent Flight Environment Close to the Ground and Its Effects on Fixed and Flapping Wings at Low Reynolds Number," *5th European Conference for Aerospace Sciences (EUCASS)*, pp. 1–10, 2013.

[22] A. Mohamed, S. Watkins, R. Clothier, and M. Abdulrahim, "Influence of Turbulence on MAV Roll Perturbations," *International Journal of Micro Air Vehicles*, vol. 6, no. 3, pp. 175–190, 2014.

[23] P. Poksawat, L. Wang, and A. Mohamed, "Automatic tuning of attitude control system for fixed-wing unmanned aerial vehicles," *IET Control Theory & Applications*, vol. 10, no. 17, pp. 2233–2242, 2016.

Collision Avoidance of multiple MAVs using a multiple Outputs to Input Saturation Technique

C. Chauffaut, L. Burlion*, F. Defaÿ and H. de Plinval
ISAE-Research, Toulouse, France
ONERA - The French Aerospace Lab, Toulouse, France

ABSTRACT

This paper proposes a novel collision avoidance scheme for MAVs. This scheme is based on the use of a recent technique which is based on the transformation of state constraints into time-varying control input saturations. Here, this technique is extended so as to ensure collision avoidance of a formation of up to three MAVs. Experimental results involving three A.R drones show the efficiency of the approach.

1 INTRODUCTION

The Unmanned Aicraft Integration into civil Airspace is a major challenge which involves considering new uses for these vehicles while reducing technical barriers related to safety. Underlying many of the safety challenges is the issue of assessing the capability of one operator to simultaneously control multiple vehicles. The need for basic formation keeping techniques and especially collision avoidance capability is surely the most critical in order to enable one operator to focus on the supervision of the fleet of Unmanned Aerial Vehicles (MAVs).

A large research effort has been focused on the synchronization and formation control of a fleet of MAVs (see e.g the survey paper [1] and the references therein). Many interesting problems have been adressed so far : to cite a few, researchers have considered formation control problems dealing with leader-follower approaches [2, 3, 4, 5], cooperative sensing [6, 7] or communication delays [8]. Another difficult problem is the collision avoidance between members of a fleet and/or the environment. This problem was mainly adressed using the well known potential field method [9, 10] which is not straightforward to apply when one considers underactuated MAVs (as discussed in the open problems section 8.3 of [11]) or when one would like to choose a sophisticated dynamical guidance laws when the MAVs are far from each other.

This paper introduces a novel anti-collision technique, different from the potential field methods : it makes use of the Output to Input Saturation Transformation (OIST) method. First presented in [12], and later applied to visual servoing [13, 14] and load alleviation for a civil aircraft [15]. The principle of this method is to transform a desired bound on a variable of interest into a saturation expressed on the control inputs. One of the main features of this approach is that a smooth switch between a nominal (local) control law and a saturated (global) one can be performed. Also, once transformed into a saturated input control problem, the formulation boils down to a well-known problem for which an abundant literature is available. Thus, the anti-windup framework [16] can be applied to the problem transformed via OIST [17]. In this paper, we present experimental results which demonstrate the successful application of this methodology to a triangular formation of MAVs.

Section 1 indroduces the problematic of obstacle avoidance and formation flying. Section 2 presents the simplified dynamic model used for the control. In Section 3, the theoretical development used to ensure safe collision avoidance based on the OIST framework is presented. Then Section 4 describes the system architecture. Section 5 describes the experimental results. Finally, conclusions follow.

2 MAV DYNAMIC MODEL FOR CONTROL

For the control synthesis, the MAV is modeled as a 3 DOF mass without taking the drag of the mav into account. The considered control input is the thrust vector. This thrust can easily be converted into a global thrust provided by the rotors velocities, and its orientation which may be obtained by differences between these velocities. As a result, from now on, we consider that the thrust vector may be chosen as if it were the control input. The equation of the dynamics is given as:

$$m\ddot{\boldsymbol{\xi}} + \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = \mathbf{F_d} \tag{1}$$

where $\boldsymbol{\xi} := [x\ ,\ y\ ,\ z]^T \in \mathbb{R}^3$ is the MAV position and where $\mathbf{F_d} := [F_{d,x}\ ,\ F_{d,y}\ ,\ F_{d,z}]^T \in \mathbb{R}^3$ is the control input.

2.1 Inner loop controller (PX4)

The experiments will be based on the framework presented in Section 4 which is done by two separate cards and control dynamics. Figure 1 describes the data flow between cards and MAV.

The attitude control inner loop is made by the pixhawk communauty, only the tunning of the gain have been tuned in the lab. Using the offboard mode, the baseline postion control combined to the OIST methodology send directly the

*Corresponding author: laurent.burlion(at)onera.fr

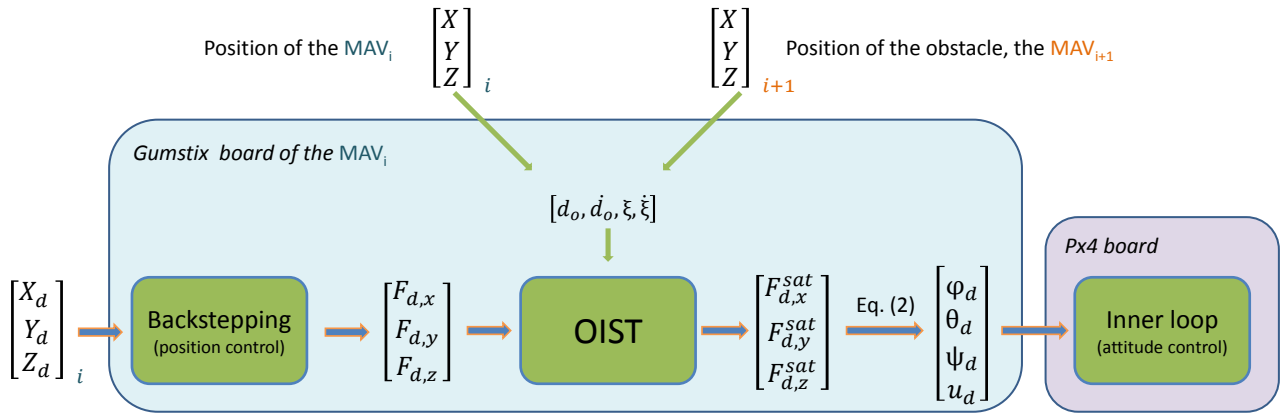


Figure 1: Output to Input Saturation Transformation methodology diagram

Attitude Target ($\varphi_d, \theta_d, \psi_d$, u_d) at the PX4 board. Figure 2 presents the actual inner loop dynamics and response. One can see the presence of static errors, and the slowness of the dynamics. These phenomena may be explained by the lack of integral term in the attitude loop of the PX4, and also by the poor capabilities of the AR drone platform – especially with the weight added for our experiments. The nominal thrust command is arround 70 percent.

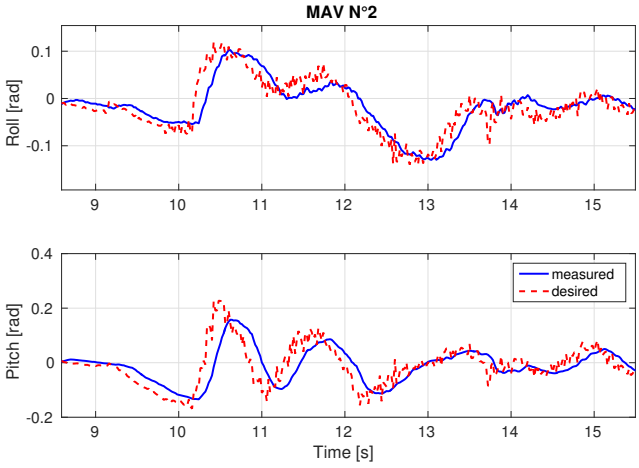


Figure 2: Attitude control dynamics

2.2 *Baseline controller (backstepping)*

The baseline controller $[F_{d,x}, F_{d,y}, F_{d,z}]$ used was developed at ISAE for trajectory tracking. More details on this control law can be found in [18].

The computed control input is expressed in terms of the thrust vector, as said earlier in this paper; it may be converted into the desired roll φ_d, pitch θ_d and thrust u_d through the simple following computations:

$$\begin{pmatrix} \varphi_d \\ \theta_d \\ u_d \end{pmatrix} = \begin{pmatrix} \text{atan2}(F_{d,y}, F_{d,z}) \\ \text{atan2}(-F_{d,x}, \sqrt{F_{d,y}^2 + F_{d,z}^2}) \\ \sqrt{F_{d,x}^2 + F_{d,y}^2 + F_{d,z}^2} \end{pmatrix} \quad (2)$$

The desired roll φ_d, pitch θ_d and thrust u_d are then transmitted as reference for the inner loop. The desired yaw ψ_d will be equal to zero.

3 OUTPUT TO INPUT SATURATION TRANSFORM (OIST) MAV AVOIDANCE

The OIST methodology applies to the aforementioned baseline controller [18], which output is saturated through the method in order to force the variable of interest to abide by the predefined constraints

3.1 *Useful notations*

Given two real numbers $x_{\min} < x_{\max}$, we note:

$$x \longmapsto Sat_{x_{\min}}^{x_{\max}}(x) = \max\left(x_{\min}, \min(x, x_{\max})\right) \quad (3)$$

the saturation function of a variable x between $x_{\min}$ and $x_{\max}$.
With an abuse of notation:

$$Sat_{x_{\min}}^{+\infty}(x) = \max(x_{\min}, x) \quad (4)$$

3.2 *OIST methodology for a moving obstacle avoidance*

Here, we propose to revisit the OIST design of [14] which was used to make a MAV avoid some obstacles. In this preliminary work, the OIST methology was successfully used for obstacle avoidance but was "switched off" (by a simple logic detailed in the experiment section of the paper) on the frontier of the obstacles in order to avoid the MAV to be stuck on them. As we shall see, an additional output constraint is here used to circumvent this problem : roughly speaking, the OIST method is not only used to avoid another

MAV but also to avoid that a MAV is stuck on another one by imposing its (orthoradial) velocity not to vanish on it. We also slightly extend the OIST framework to take into account the fact that the obstacle (other MAV) can now move (in the same (x,y) plane as the MAV), which requires to measure the velocity and acceleration of the obstacle.

The following assumptions are necessary to apply this novel OIST technique.
At any time:

- the distance d_o between the MAV and the other MAV is measured.
- the obstacle is supposed to be included inside a circle which is centered on (x_o, y_o, z) and whose radius is always lower than $d_{o,inf}$ in the (x, y) plane The obstacle being possibly in motion, its velocity (resp. acceleration) vector $(\dot{x}_o, \dot{y}_o)$ (resp. $(\ddot{x}_o, \ddot{y}_o)$) are supposed to be measured in the (x, y) plane.
- the MAV desired position $\boldsymbol{\xi_d}$ is sufficiently far from the obstacle so that the mission is feasible.

Figure 1 represents the general principle of the OIST methodology when extended to the MAV avoidance problem.
Let us define the following quantities:

$$d_{o,2} := d_o^2 = (x - x_o)^2 + (y - y_o)^2 \tag{5}$$

and

$$\phi_\perp = -(y - y_o)v_x + (x - x_o)v_y \tag{6}$$

To avoid a collision, the following constraint must be satisfied:

$$d_{o,2} \geq d_{o,inf}^2 \tag{7}$$

Moreover, to avoid staying stuck on the obstacle boundary, we consider the following additional constraint:

$$\phi_\perp \geq d_{o,inf} v_\perp^\# - \kappa_3 (d_{o,2} - d_{o,inf}^2) \tag{8}$$

where $v_\perp^\# > 0, \kappa_3 > 0$

Such an inequality simply expresses the constraint that the orthoradial velocity (component of the MAV velocity which is perpendicular to a line relating the MAV to the obstacle center) cannot be null on the obstacle boundary so that the MAV is forced to keep turning around the obstacle.

Remark 1 *another possibility is to use the constraint:*

$$\phi_\perp \leq -d_{o,inf} v_\perp^\# + \kappa_3 (d_{o,2} - d_{o,inf}^2) \tag{9}$$

where $v_\perp^\# > 0, \kappa_3 > 0$. This would change the sign of $v_\perp$ on the obstacle. This other possiblity can be considered as another degree of freedom of the method which deserves to be studied in the future.

Following the OIST methodology (see e.g. the guidelines of [13], subsection II-D)), we compute the successive time derivatives of the constrained outputs $d_{o,2}$ and $\phi_\perp$ till the input terms appear. Deriving two times (resp. one time) $d_{o,2}$ (resp. $\phi_\perp$), we obtain:

$$\begin{cases} \ddot{d}_{o,2} = 2\Big((x - x_o)\left(\frac{F_{d,x}}{m} - \ddot{x}_o\right) + (y - y_o)\left(\frac{F_{d,y}}{m} - \ddot{y}_o\right) \\ \qquad +(\dot{x} - \dot{x}_o)^2 + (\dot{y} - \dot{y}_o)^2\Big) \\ \dot{\phi}_\perp = -(y - y_o)\frac{F_{d,x}}{m} + (x - x_o)\frac{F_{d,y}}{m} - (\dot{y} - \dot{y}_o)v_x \\ \qquad +(\dot{x} - \dot{x}_o)v_y \end{cases} \tag{10}$$

We now consider the following matrix:

$$\mathbf{M_o}(\boldsymbol{\xi}) := \frac{2}{m}\begin{bmatrix} x - x_o & y - y_o \\ -(y - y_o) & x - x_o \end{bmatrix} \tag{11}$$

which is invertible when $det(\mathbf{M_o}) = d_{o,2} \geq d_{o,inf}^2 > 0$.
Using $\mathbf{M_o}$, we rewrite (10) as follows:

$$\begin{bmatrix} \ddot{d}_{o,2} \\ 2\dot{\phi}_\perp \end{bmatrix} = \mathbf{M_o}(\boldsymbol{\xi}) \begin{bmatrix} F_{d,x} \\ F_{d,y} \end{bmatrix} + \begin{bmatrix} D_1 \\ 2D_2 \end{bmatrix} \tag{12}$$

where

$$\begin{aligned} D_1 &= 2(\dot{x} - \dot{x}_o)^2 + 2(\dot{y} - \dot{y}_o)^2 - 2(x - x_o)\ddot{x}_o \\ &\quad -2(y - y_o)\ddot{y}_o \qquad (13) \\ D_2 &= -(\dot{y} - \dot{y}_o)v_x + (\dot{x} - \dot{x}_o)v_y \qquad (14) \end{aligned}$$

$\mathbf{M_o}$ being invertible when the first constraint is respected, we define the following change of coordinates:

$$\begin{bmatrix} u_1 \\ 2u_2 \end{bmatrix} := \mathbf{M_o}(\boldsymbol{\xi}) \begin{bmatrix} F_{d,x} \\ F_{d,y} \end{bmatrix} \tag{15}$$

We then rewrite (12) as follows:

$$\begin{bmatrix} \ddot{d}_{o,2} \\ \dot{\phi}_\perp \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} \tag{16}$$

Let us now define $\dot{d}_{o,2} := \frac{d}{dt} d_{0,2} = 2 d_o \dot{d}_o$ and the OIST tuning parameters $\kappa_2, \kappa_3, \kappa_4 > 0$.
We've got the following result:

Proposition 1 *Let us suppose that $d_{o,2}(t = 0) \geq d_{o,inf}^2$, $\dot{d}_{o,2}(t = 0) \geq -\kappa_1(d_{o,2}(t = 0) - d_{o,inf}^2)$ and $\phi_\perp(t = 0) \geq d_{o,inf} v_\perp^\# - \kappa_3(d_{o,2}(t = 0) - d_{o,inf}^2)$, then if for all $t \geq 0$,*

$$\begin{aligned} u_1 &\geq -(\kappa_1 + \kappa_2)\dot{d}_{o,2} - \kappa_1\kappa_2(d_{o,2} - d_{o,inf}^2) - D_1 \qquad (17) \\ u_2 &\geq -\kappa_3 \dot{d}_{o,2} - \kappa_4(\phi_\perp - d_{o,inf} v_\perp^\# + \kappa_3(d_{o,2} - d_{o,inf}^2)) \\ &\quad -D_2 \qquad (18) \end{aligned}$$

then the output constraints (7)-(8) are satisfied for all $t \geq 0$.

proof: Straightforward applying Lemma 4.1 of [17].

Finally, it is required to express the input constraints in terms of the original control inputs.
It is easy to see that (17)-(18) are satisfied when one applies the following inputs:

$$\begin{bmatrix} F_{d,x}^{sat} \\ F_{d,y}^{sat} \end{bmatrix} = \mathbf{M_o}(\boldsymbol{\xi})^{-1} \begin{bmatrix} Sat^{+\infty}_{h_1(\dot{\xi},d_o,\dot{d}_0)}(u_1) \\ 2Sat^{+\infty}_{h_2(\dot{\xi},d_o,\dot{d}_0)}(u_2) \end{bmatrix} \tag{19}$$

where

$$\begin{aligned} h_1(\dot{\boldsymbol{\xi}}, d_o, \dot{d}_0) = &-D_1 - (\kappa_1 + \kappa_2)\dot{d}_{o,2} \\ &-\kappa_1\kappa_2(d_{o,2} - d_{o,inf}^2) \end{aligned} \tag{20}$$

$$\begin{aligned} h_2(\dot{\boldsymbol{\xi}}, d_o, \dot{d}_0) = &-D_2 - \kappa_3\dot{d}_{o,2} - \kappa_4(\phi_\perp - d_{o,inf}v_\perp^{\#}) \\ &-\kappa_3\kappa_4(d_{o,2} - d_{o,inf}^2)) \end{aligned} \tag{21}$$

Using (15), we finally obtain the following input saturations with time varying bounds :

$$\begin{bmatrix} F_{d,x}^{sat} \\ F_{d,y}^{sat} \end{bmatrix} \tag{22}$$

$$= \mathbf{M_o}(\boldsymbol{\xi})^{-1} \begin{bmatrix} Sat^{+\infty}_{h_1(\dot{\xi},d_o,\dot{d}_0)} \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}^T \mathbf{M_o}(\boldsymbol{\xi}) \begin{bmatrix} F_{d,x} \\ F_{d,y} \end{bmatrix} \right) \\ 2Sat^{+\infty}_{h_2(\dot{\xi},d_o,\dot{d}_0)} \left(\frac{1}{2} \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T \mathbf{M_o}(\boldsymbol{\xi}) \begin{bmatrix} F_{d,x} \\ F_{d,y} \end{bmatrix} \right) \end{bmatrix} \tag{23}$$

4 SYSTEM ARCHITECTURE

4.1 Framework for experimentation

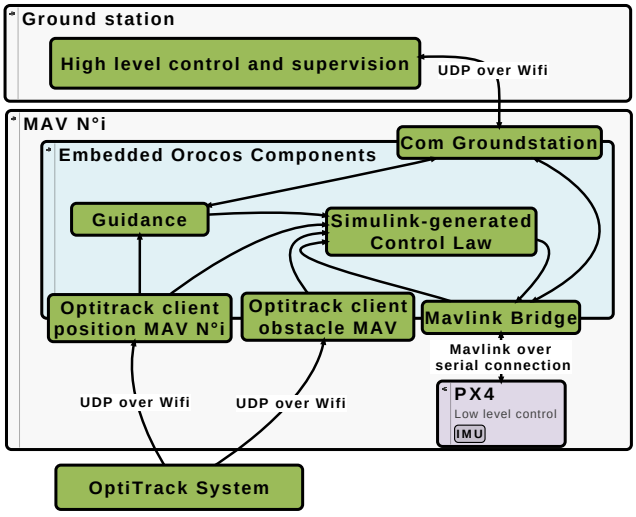


Figure 3: OIST formation framework

The framework that is used in our lab is described in [14] and allows many use cases, from the simple Simulink-only simulation to the experiments involving interaction between a real MAV and simulated sensors in the MORSE simulator. A variety of MAVs are available for experiments in our lab. For the formation flying experiment, the framework used (Fig. 3) was:

1. **Ground station:**

   A ground station was used to transmit high-level orders to the 3 MAVs using Wi-Fi. The high-level orders were "take off", "move formation to position $[x, y, z]$", "land", and "arm/disarm motors". It also monitored the battery level and position of the MAVs.

2. **Embedded Orocos components:**

   An Orocos component was used to relay the high-level orders received from the ground station to the guidance ("take off", "move to" and "land") and mavlink bridge ("arm/disarm motors") components. The MAV's desired position transmitted to the guidance component was offset to reflect the MAV N°i position in the formation. From the desired position, the guidance component was generating desired trajectories for positions and velocities without taking into account the obstacle. These trajectories were given to the Simulink-generated Orocos component which implemented the OIST control law. The OptiTrack was used to measure the MAV and MAV's obstacle positions at 50Hz.

3. **MAV N°$\mathbf{i}, \mathbf{i} \in \{0, 1, 2\}$:**

   The MAVs used were based on the mechanics of three Parrot AR.drone with custom electronics, a gumstix running the Orocos components and a Pixhawks PX4 for the attitude control.

4.2 Code-generation from Simulink models

Automatic code generation is used directly from Simulink to generate C++ Code which is included in the framework.

5 EXPERIMENT

In order to demonstrate the OIST obstacle avoidance methodology, we use it for the anti-collision of a fleet composed of three MAVs. The three MAVs take off on the ground at three different positions, then they go to their target point in the formation pattern. For i=0 to 2, each MAV N°i considers the MAV N°$i+1$ as a moving obstacle and will avoid it.(By convention MAV N°3 is MAV N°0 so that MAV N°2 avoids MAV N°0).

Figure 4 presents the trajectory of the MAVs with obstacle avoidance during the formation Establishment. A minimal distance of 80 centimeters has been chosen for the avoidance parameter.

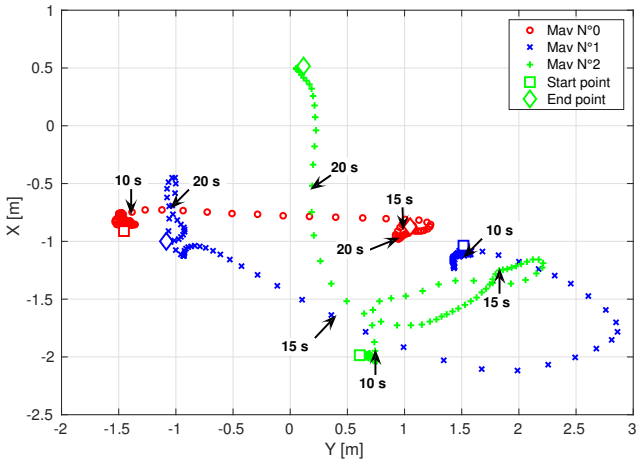


Figure 4: Trajectory of the 3 MAV's during the flight

One can see that the MAV N°0 is not deviated from the desired trajectory given by the guidance component. The two other MAVs are deviated by OIST as denoted on Fig. 5.

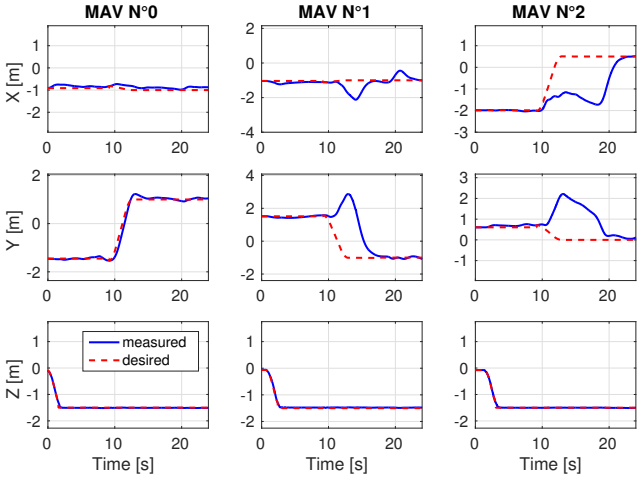


Figure 5: Positions of the 3 MAV's during the flight, the desired trajectories are computed without taking into account the obstacles

The avoidance results can be see on Fig. 6 which shows the saturation of the baseline thrust F_d by the OIST methodology, and the distance between the MAV N°i and its obstacle (the position of the MAV N°$i+1$).

For this experiment, the A.R.drone frame doesn't allow to increase the tracking performance of the position due to the lack of power. In our lab, the MAV based on Parrot weight more than 500 grams, which is 20 percent more than the nominal weight of the A.R.drone, so the thrust at the equilibrium point (for static flight) is about 75%. The dynamic cannot be very fast and we need to increase the minimal distance to 80 centimeters. On Fig. 6, the MAV N°2 is inside the avoidance

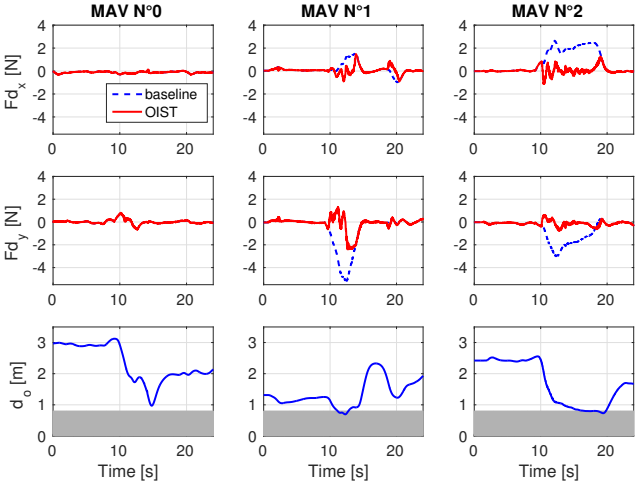


Figure 6: Saturation of the baseline thrust F_d and distance to obstacles during the flight

interval but as it is shown on Fig. 5, the tracking error is not minimal (about 20 centimeters of error) so this is more due to the position control.

In the video of this experiment [19], we can see that the MAVs avoid each other during the establishment of the formation. At the second 34 in the video (second 18 in the Figs. 4 to 6), we can see that even if the distance between the MAVs N°1 and N°2 is greater than 80 cm, the MAV N°1 still moves out of the way. This is because the OIST methodology also takes into account the velocity of the obstacle (MAV N°2 in this case). This effect can also be seen on the MAV N°2 between the seconds 10 to 12 when the MAV N°0 is approaching quickly.

6 CONCLUSIONS

In this paper, a novel framework to ensure anti-collision of a fleet of three MAVs has been presented. The OIST methology has been extended so that a MAV cannot be stuck on another MAV but turns around it. The main feature of this method is its simplicity in terms of control design and computational load. An experiment has been performed and validates the OIST methodology to ensure collision avoidance of up to three MAVs. The experimental results shows that the minimal distance between MAVs is globally respected during the establishment of the formation of the fleet. For the future experiments, MikroKopter-based MAVs will be used to increase the dynamics and the capabilities of the OIST methodology. Future works include rigourus stability proofs of saturated systems (in the spirit of [17]), extra experiments involving a larger number of MAVs and the extension of the OIST technique to address other problems related to safe guidance of multiple MAVs.

REFERENCES

[1] Y. Zhang and H. Mehrjerdi. A survey on multiple unmanned vehicles formation control and coordination: Normal and fault situations. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1087–1096, 2013.

[2] B. D. O. Anderson, B. Fidan, C. Yu, and D. Walle. *UAV Formation Control: Theory and Application*. Springer London, 2008.

[3] F. Giulietti, L. Pollini, and M. Innocenti. Autonomous formation flight. *IEEE Control Systems*, 20(6):34–44, 2000.

[4] Y. Watanabe, A. Amiez, and P. Chavent. Fully-autonomous coordinated flight of multiple uavs using decentralized virtual leader approach. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5736–5741, 2013.

[5] L. V. Santana, A. S. Brando, and M. Sarcinelli-Filho. Heterogeneous leader-follower formation based on kinematic models. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 342–346, 2016.

[6] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson. Decentralized cooperative aerial surveillance using fixed-wing miniature uavs. *Proceedings of the IEEE*, 94(7):1306–1324, 2006.

[7] G. Gu, P. R. Chandler, C. J. Schumacher, A. Sparks, and M. Pachter. Optimal cooperative sensing using a team of uavs. *IEEE Transactions on Aerospace and Electronic Systems*, 42(4):1446–1458, 2006.

[8] A. Seuret, D. V. Dimarogonas, and K. H. Johansson. Consensus of double integrator multi-agents under communication delay. *FAC Proceedings Volumes*, 42(14):376 – 381, 2009. 8th IFAC Workshop on Time-Delay Systems.

[9] R. Olfati-Saber and R.M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. *FAC Proceedings Volumes*, 35(1):495 – 500, 2002. 15th IFAC World Congress.

[10] L. Garcia-Delgado, A. Dzul, V. Santibanez, and M. Llama. Quad-rotors formation based on potential functions with obstacle avoidance. *IET Control Theory Applications*, 6(12):1787–1802, 2012.

[11] A. Abdessameud and A. Tayebi. *Motion Coordination for VTOL Unmanned Aerial Vehicles*. Springer, 2013.

[12] L. Burlion. A new saturation function to convert an output constraint into an input constraint. In *Control Automation (MED), 2012 20th Mediterranean Conference on*, pages 1217–1222, 2012.

[13] L. Burlion and H. de Plinval. Keeping a ground point in the camera field of view of a landing uav. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5763–5768, 2013.

[14] C. Chauffaut, F. Defay, L. Burlion, and H. de Plinval. Uav obstacle avoidance scheme using an output to input saturation transformation technique. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 227–234, 2016.

[15] L. Burlion, C. Poussot-Vassal, P. Vuillemin, M. Leitner, and T. Kier. Longitudinal manoeuvre load control of a flexible large-scale aircraft. In *the 19th IFAC World Congress Conference*, pages 3413–3418, 2014.

[16] S. Galeani, S. Tarbouriech, M. Turner, and L. Zaccarian. A tutorial on modern anti-windup design. *European Journal of Control*, 15(34):418 – 440, 2009.

[17] E. Chambon, L. Burlion, and P. Apkarian. Time-response shaping using output to input saturation transformation. *International Journal of Control*, to appear in 2017.

[18] M. Lecointe, F. Defaÿ, and C. Carvalho Chanel. Backstepping control law application to path tracking with an indoor quadrotor. In *3rd CEAS EUROGNC Conf.*, 2015.

[19] C. Corentin, F. Defaÿ, L. Burlion, and H. de Plinval. Video of the oist experiment for collision avoidance in a fleet of three mavs, http://video.isae.fr/videos/?video=media170215164039955, 2017.

A Combined Approach for 3D Formation Control in a Multi-UAV System using ROS

Rafael G. Braga[(a)]*, Roberto C. da Silva[(a)], Alexandre C. B. Ramos[(a)] and Felix Mora-Camino[(b)]
[(a)]Institute of Mathematics and Computer Science, Federal University of Itajubá, Itajubá, MG, Brazil
[(b)]Air Transport Department, Ecole Nationale de l'Aviation Civile, Toulouse, France

ABSTRACT

Formation control in multi-UAV systems can be obtained through different strategies, each one with its own advantages and disadvantages. In order to minimize the weaknesses of each technique, this paper proposes a combined approach to drive a group of three quadrotor UAVs in a time varying formation, using a virtual structure, a leader-follower strategy and two behavioral rules. To each UAV is assigned a position in a formation that is represented by a virtual structure. The UAVs then have to compute their desired positions in order to achieve the formation. This is done using one of two possible methods, one based on a leader-follower approach and another based on waypoints received from a ground station. Two behavioral rules are then used to move the UAVs towards their goal while avoiding collisions with each other. The algorithm was implemented in C++ using the ROS platform and was tested in simulations using the Pixhawk SITL simulator. Results show that the UAVs are able to move in formation and also to change the formation without colliding with each other.

1 INTRODUCTION

Recently there have being a growing interest in the use of UAVs (Unmanned Aerial Vehicles) in a wide range of applications. This is due to the development of inexpensive and easy to build UAV models that can carry powerful sensors and even miniature computers. The number of proposed applications is large, ranging from civil uses such as forest fire monitoring and fighting [1][2], buildings and bridges inspection [3], crop dusting [4] and search and rescue of survivors after a disaster [5], to millitary uses including surveillance and monitoring of an area [6] or even air strikes [7]. One of the most popular model both in commercial use and in academic research is the quadrotor, thanks to its simplicity and great mobility.

Since the control of one UAV is already well understood, a new problem that is attracting attention is the use of a group of UAVs to perform missions. There are many advantages that come from this approach: a group carries more sensors and so is able to acquire more information about the environment; the number of UAVs in the swarm can be altered to tackle missions of different levels of difficulty; if one UAV fails the remaining ones can continue the mission without it. All these advantages however come with the drawback of being more difficult to control a swarm correctly than a single unit.

There are many examples of successful attempts in controlling groups of UAVs, using techniques proposed in the swarm robotics field. Kushleyev et al. developed an impressive flock of 20 minature quadrotors capable of assuming many different formations, using a centralized control strategy [8]. Bürkle et al. created an outdoor quadcopter swarm also using central processing at a ground station [9]. Vásárhelyi et al. used flocking rules to fly a remarkable swarm of 10 quadrotors in an outdoor environment [10]. And in [11] a group of simulated UAVs is used to monitor an area.

Most of the control algorithms proposed in the literature can be classified as three types: leader-follower [12][13], virtual structure [14][15] or behavior based [16][17]. In the leader-follower strategy, one of the UAVs in the group is chosen as the leader. The others have to follow it and position themselves according to its position. This approach has the advantage of being easier for a human operator to drive the group, having to control only the leader. However one disadvantage is that if the leader stops working, the whole group also stop. The virtual structure strategy consists in treating the whole group as a single fixed structure, with each UAV representing one point that composes it. The controller is designed such as each UAV is moved to create the structure's desired behavior. Finally, in the behavior-based strategy the UAVs are programmed to follow some desired behavior, such as avoiding collisions or move closer to one another. In most of the cases, these kind of strategy is based in real phenomena observed in nature, and so is classified as bio-inspired.

To control a group of three UAVs while avoiding collisions with each other and keeping track of a time varying formation, we propose an approach that combines elements from the three mentioned strategies. The formation is treated as a rigid structure, composed by an array of poses that will be assigned to each UAV in the group. The leader is the only robot that can move freely, being controlled by an operator at a ground station or navigating autonomously. All the other

*Email address(es): rafaelbraga@unifei.edu.br

UAVs will try to move autonomously to their assigned poses in the formation, relative to the leader. The position control is behavior based, using two behavioral rules to move the UAVs: formation and separation.

The rest of this paper is structured as follow. Section 2 presents our multi-UAV system and describes our hardware and software platforms. Section 3 describes the design of the formation control strategy and the pseudocode implementation of the behavioral rules. Finally, Section 4 presents the simulation environment and the obtained results.

2 Materials and Methods

2.1 Multi-UAV System

Our system consists of a group of robots that fly over an area and a ground station with which they communicate through a telemetry link. The ground station is responsible for collecting flight data from each robot and sending information about the desired formation, but not controlling them. The control algorithm is distributed and runs in each UAV. The objective is to move the group to a desired point, avoiding collisions between the UAVs and keeping a formation that can be changed over time.

The quadrotor is a simple machine, capable of vertical take-off and landing (VTOL) and moving with six Degrees of Freedom (DoF). It consists of a center body with four individual rotors attached, as illustrated in Figure 1. By controlling the thrust generated by each rotor, we can lift the quadrotor and move it in the air. As shown in Figure 1, rotor i rotates anticlockwise if i is even and clockwise if i is odd. By adjusting the speed of the clockwise and anticlockwise rotors we can control the Yaw angle.

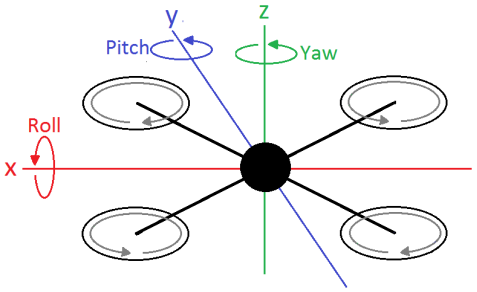


Figure 1: Degrees of freedom achieved by the quadrotor and rotation direction of the rotors

2.2 UAV Hardware

The robots we are developing the algorithm to control are small quadrotors with 250mm diameter, each one using a Pixhawk as a flight controller board. The Pixhawk is an open-source device [18] equipped with many sensors such as gyroscope, accelerometers, magnetometer and barometer and can also be connected to a external GPS module. It runs a powerful software that implements the basic controller routines of the quadrotor, along with many other useful functions. In our application, the main function of the Pixhawk is to provide the low-level stabilization and height control of the UAV. Figure 2 shows a photo of one of the UAVs used in our laboratory.

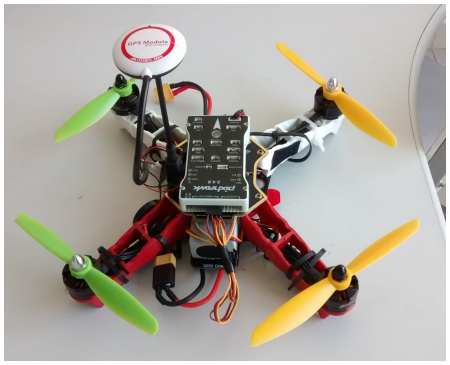


Figure 2: UAV used in the laboratory

Originally the Pixhawk is intended to be controlled by a human operator via radio controller or receive commands from a ground station. However, it is also able to communicate with other devices via a protocol called MAVLink. We use this protocol to send commands from an embedded Linux computer (Raspberry Pi) which is also carried by the UAV. in this way we can program the UAV to fly autonomously, while still being able to regain manual control at any time. Figure 3 shows a schematic view of the UAV components.

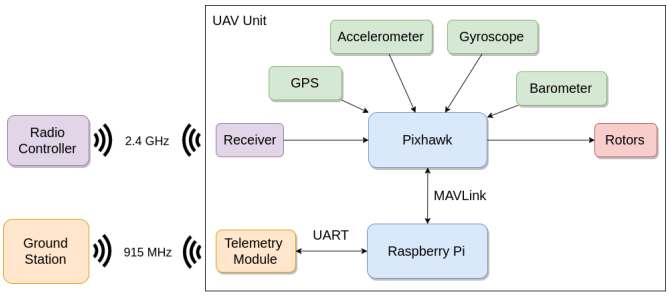


Figure 3: UAV components

2.3 Robot Operating System

Our algorithm was implemented in C++ and runs on the ROS platform. ROS (Robot Operating System) is an open source framework created to aid researchers in developing robotic applications [19]. ROS provides us with many tools and facilities that were very useful in our work.

A ROS application is a network of independent programs, called nodes, that communicate with each other. This network is managed by another program called ROS Master. The communication works in a message passing way. Nodes that generate data publish this information in topics in the form

of messages, while nodes that need that data subscribe to the corresponding topics and receive the published messages. The types of messages represent common data structures used in robotics, such as sensor readings or velocity commands.

In our application, the ROS system, containing all the control algorithms, runs on the Raspberry Pi. We used a node called mavros which connects to the Pixhawk via a serial connection and is able to translate MAVLink messages into ROS messages and vice versa. In this way the control node can get data from the Pixhawk and send commands to it. This architecture is represented in Figure 4.

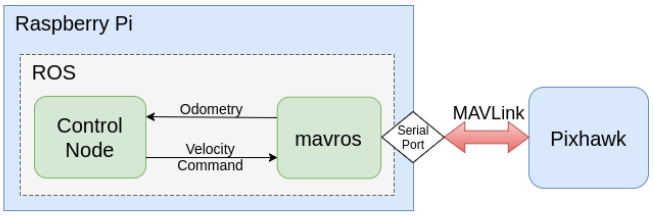


Figure 4: System architecture overview

3 Development

3.1 Formation Control Strategy

We define a global reference frame S_G with X, Y and Z Cartesian coordinates, fixed on the ground. The position of each UAV i in the S_G frame is given by:

$$p_i^G = (X_i^G, Y_i^G, Z_i^G) \tag{1}$$

Each robot uses its embedded sensors to locate itself in the global frame. Considering the Earth's geometry and the origin of the S_G frame, the latitude and longitude information from the GPS sensor can be converted to values in meters in the X and Y coordinates. This gives us the robot's initial position. From this moment on, the position will be obtained from the Pixhawk's internal state estimator, which uses an Extend Kalman Filter to fuse the measurements from its embedded sensors, including gyroscope and accelerometers. The Z coordinate is obtained from the GPS and the barometer information fused together.

Each UAV receives an unique ID, which is a positive integer number starting at 0 and increasing in increments of 1. The formation is defined as an array of positions relative to a formation reference frame, S_F. The assigned position to each UAV is the one whose index in the formation array is equal to the UAV's ID. In our experiments we defined three formations: a horizontal line, a vertical column and a triangle.

Each UAV has to compute its desired position relative to the formation frame. We defined two methods to make this calculation:

- **Leader-follower method:** In this method, one UAV is considered the leader of the group (the one with ID 0). The leader ignores its position in the formation and instead is able to move freely, being controlled by an operator or following a sequence of waypoints. The other UAVs calculate their desired position relative to the leader, considering it the origin of the formation frame.
- **Waypoint method:** In this method, all UAVs receive a waypoint to follow and treat it as the origin of the formation frame. There is no leader in this method. As such, this approach is more tolerant to UAV failure than the first one, however is more difficult for a human operator to drive the group.

The position of each UAV i in the S_F frame is represented by p_i^F. In the first position calculation method, the origin of the S_F frame is the position of the leader, and then the desired position of each UAV i in the global frame is given by:

$$p_{iD}^G = p_0^G + p_i^F \tag{2}$$

where p_{iD}^G is the desired position of UAV i in the S_G frame and p_0^G is the leader's current position in the S_G frame. In the second method, the S_F frame is centralized at the next waypoint, so the desired position of UAV i in the global frame is given by:

$$p_{iD}^G = p_w^G + p_i^F \tag{3}$$

where p_w^G is the position of the next waypoint.

3.2 The ROS Application

Two nodes are executed by each UAV: mavros and the control node, called formation_controller_node, which is our main control program. This node subscribes to some of the topics where mavros publishes data received from the Pixhawk. Using information from the GPS sensor and the internal state estimator of the Pixhawk, the formation_controller_node determines the position of the UAV in the global reference frame. It then publishes this information in a topic called /uav_positions. The node also subscribes to this same topic, but since all UAVs are publishing their positions in this topic, the node receives the position information of all other UAVs. This data will be used by the behavioral rules to move the UAV.

The formation_controller_node also subscribes to a topic called /formation, where the ground station publishes messages of type geometry_msgs/PoseArray. These messages contain an array of positions representing the formation virtual structure. Each time a new message arrives, the formation_controller_node updates an internal formation array variable, which means that the formation can be changed during the flight.

Using all the information obtained, the formation_controller_node then uses the behavioural rules to drive the UAV by publishing a geometry_msgs/TwistStamped

message in the mavros/setpoint_velocity/cmd_vel topic. The mavros node receives this message, creates the corresponding MAVLink message and then sends it to the Pixhawk, which will follow that command accordingly.

Since each UAV runs instances of the same nodes, it is necessary to run them under different namespaces to avoid causing conflicts. To each UAV is assigned a namespace in the form of /uavn where n is that UAV's ID. A simplified diagram of the resulting system is shown in figure 5.

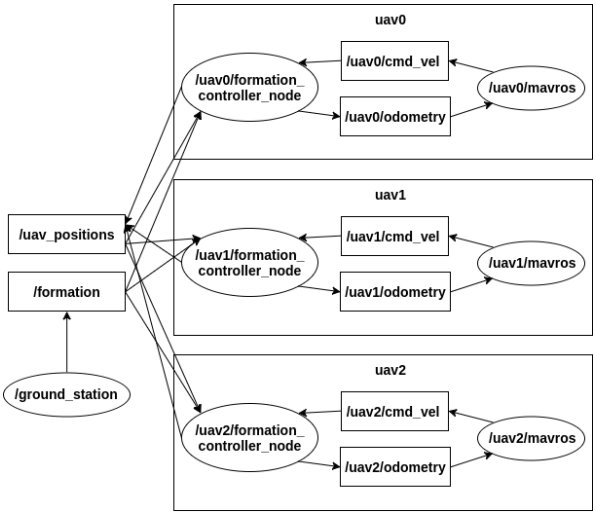


Figure 5: Simplified diagram of the ROS application generated by the program rqt_graph.

3.3 Algorithm Implementation

In this section we will describe how we implemented the two behavioral rules that move the UAVs together in a formation.

In each loop of the control algorithm, it follows the following steps: *i.* Obtain the UAV's own pose from the Pixhawk's internal state estimator; *ii.* Determine the position of the neighbors based on the position messages received from the other UAVs; *iii.* Obtain the formation array from the /formation topic; *iv.* Use the behavioural rules for Cohesion and Separation to determine the direction which the UAV should move. The main loop of the algorithm is given in Algorithm 1.

First the program calls the function *getPose()* which calculates the pose of the UAV and sets the variable *this.position* representing the UAV's position in the global frame. Then, the program calls another function *getNeighbors()* that gets the positions of the nearby robots from the received messages and stores this data in an array called *neighbors*.

Then the program starts calling the functions where the actual behavioral rules are implemented. Each function returns a vector that represent the direction that rule tells the UAV to go. For example, the Separation rule urges the UAV

Algorithm 1 Main loop of the formation control algorithm

```
1: procedure SWARM
2:     getPose()
3:     getNeighbors()
4:     getFormation()
5:     v1 ← separation()
6:     v2 ← cohesion()
7:     vres ← r1 ∗ v1 + r2 ∗ v2
8:     move(vres)
9: end procedure
```

to move away from its neighbors, so the *separation()* function will return a vector pointing to the opposite direction of the other UAVs. Each function will be explained in detail ahead.

In line 7 the two vectors are combined trough a weighted sum to generate the final resulting direction the robot should move. The weight applied to each rule determines how much that rule influences the final result. The values of the weights can be changed to obtain different results. Rules can be even completely removed from the calculation by setting its weight to zero.

In line 8 the calculated resulting direction is passed to another function that moves the UAV. This function publishes velocity commands to the mavros package, which in turn send MAVLink messages to the flight controller board, instructing it to move the UAV.

The implementation of each rule and its corresponding function will be explained in detail:

Separation Rule: This rule urges the robot to move away from other robots to avoid collisions. This is done creating a vector for each one of the detected neighbors pointing to the exactly opposite direction of that neighbor. These vectors are then combined into a resulting vector and returned to the main program. The implementation is shown in Algorithm 2.

Algorithm 2 Separation Rule

```
1: procedure SEPARATION
2:     Vector v ← 0
3:     for all neighbors n do
4:         vn ← this.position − n.position
5:         vn.normalize()
6:         vn ← vn ∗ distance(this, n)
7:         v ← v − n
8:     end for
9:     return v
10: end procedure
```

We also want the robot to move faster the closer it is from its neighbor. We do this by adjusting the vector's magnitude. First we normalize it so it becomes a unit vector. Then we divide it by the distance between the two robots. As the distance between two robots gets smaller, the magnitude of the

resulting vector becomes bigger.

Formation Rule: This rule tries to move the robot to its desired position in order to maintain the formation. The implementation is shown in Algorithm 3. Fist we test which method is being used to determine the desired position. Then we check if the UAV is the leader by testing if its ID is 0. If the method being used is the leader-follower, the leader follows the waypoint and the followers determine their desired position according to Equation 2. If the waypoint method is being used, all UAVs use Equation 3 to determine their desired position. The function then generates a vector pointing to the calculated position.

Algorithm 3 Formation Rule

```
1:  procedure FORMATION
2:      Vector v ← 0
3:      Vector DesiredPosition ← 0
4:      if leader-follower then
5:          if ID == 0 then
6:              DesiredPosition ← waypoint
7:          else
8:              DesiredPosition ← leader.position +
    Formation[ID]
9:          end if
10:     else
11:         DesiredPosition ← waypoint + Formation[ID]
12:     end if
13:     v ← DesiredPosition − this.position
14:     return v
15: end procedure
```

3.4 Computational Complexity and Communication issues

One important concern when designing a multi-robot control application is the elevation of the computational complexity with the increase in the number of robots. In special, in a behavior based approach such as the one we used, each individual has to iterate through all other individuals to compute its behavioral rules. If a centralized control is used, this represent a $O(n^2)$ computational complexity, where n is the number of robots. However, in a decentralized approach, each robot runs its own control algorithm with a computational complexity of O(n). This means that a decentralized approach can make the system more scalable, with the addition of more robots bringing a smaller burden to the process.

Another concern is the influence of delayed communications. In our application the robots must communicate with one another to be able to locate their neighbors. Since we performed our tests in simulation, the communication delay was not a problem. However, in real life implementations the system would be negatively influenced by this delay. In Section 5 we discuss how this problem could be tackled in future works.

4 Results and Discussion

We decided to run our program in simulation first as a proof-of-concept and also as a way to evaluate its performance. We used the Pixhawk SITL (Software In The Loop) simulator, a software that is provided as part of the Ardupilot project. With it we were able to interact with a simulated Pixhawk running its original firmware and generating accelerometer, gyroscope, GPS and other sensors data. Three instances of the simulator were executed simultaneously, each one using a different GPS starting position, effectively simulating three UAVs that were able to be controlled individually by our ROS application.

We defined a scenario where the three UAVs, with IDs of 0, 1, and 2, have to move together maintaining a formation. The UAVs were commanded to take off to an altitude of 2 meters. As soon as they are stabilized in this altitude, we started the control node and they start moving. Since the SITL does not generate a graphical representation of the simulation, we used the program RViz, which is a standard ROS tool to create a visual representation of the UAVs. Figure 6 shows the three UAVs in RViz screen.

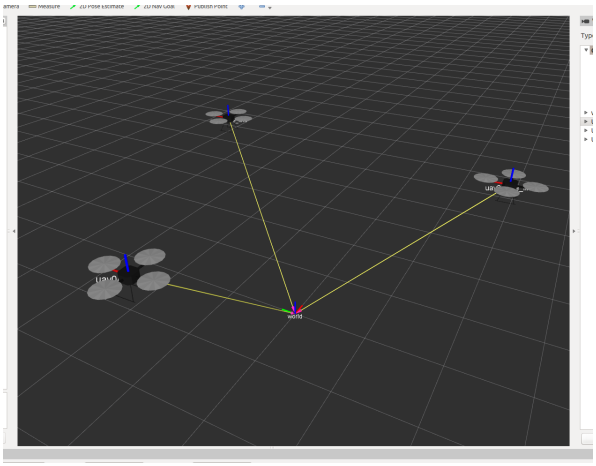


Figure 6: Visualization of simulated UAVs in RViz.

To evaluate the performance of our algorithm we defined two main tests.

4.1 Moving in Formation

The first test aims to evaluate the capability of the UAVs to maintain a formation while moving together. The robots are commanded to assume a formation and then move to two waypoints. We repeated this test for each position calculation method and for three different formations: a horizontal line, a vertical column and a triangle. In all experiments the UAVs were capable of maintaining the formation and no collision occurred. In Figure 7 we show the trajectory developed by the UAVs in the test with the triangle formation and leader-follower position calculation method.

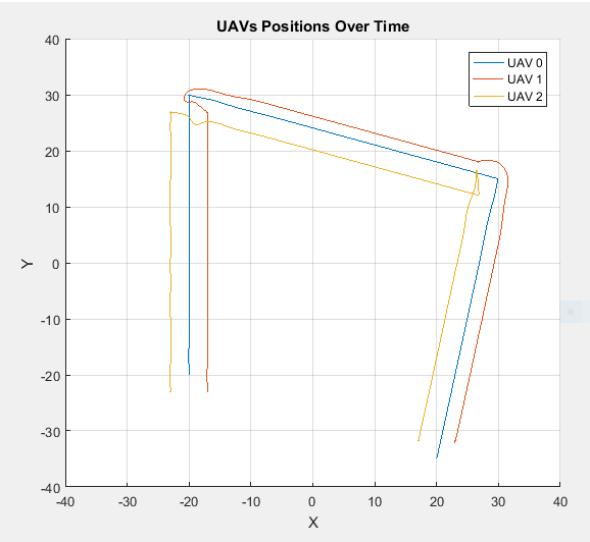


Figure 7: UAVs trajectory over time.

Another interesting feature of our solution is the possibility of formation control in three dimensions. In figure 8 the three UAVs are keeping a vertical formation, assuming different positions in the Z axis.

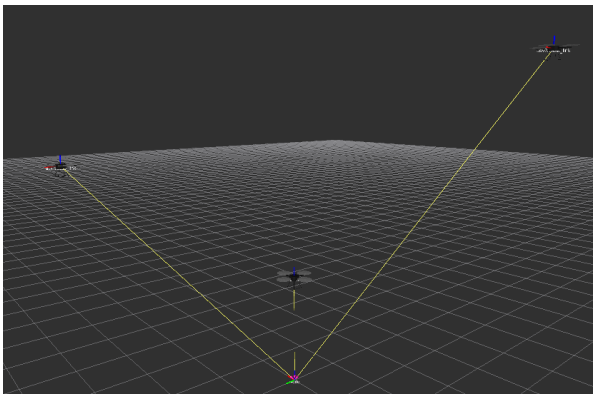


Figure 8: UAVs performing a vertical formation.

4.2 *Changing Formations*

In this test we tried to see if the UAVs were capable of changing from one formation to another without colliding with each other. The UAVs started at the line formation, then changed to other formations following this sequence: column, triangle, line, triangle, column and then line. Table 1 shows the time spent in each formation change. Thanks to the separation behavior, no collision was observed.

It is interesting to note that the longer times appear in the transitions to the column formation. This is caused by the fact that this formation is actually vertical, so the UAVs have to gain height in order to assume their desired positions.

Transition	**Time [s]**
line → column	7.16
line → triangle	3.66
column → triangle	4.90
column → line	4.70
triangle → line	3.53
triangle → column	7.17

Table 1: Time spent in each formation transition.

5 Conclusion

This paper presented the design of a combined approach to the formation control problem in a group of UAV vehicles. Our algorithm combined elements from three different multi-UAV control strategies: leader-follower, virtual structure and behaviour-based. The implementation was tested in simulations and presented good results. The UAVs were able to move in different formations and change from one formation to another without colliding with each other.

In future works some improvements can be studied, such as the use of cameras or range sensors to detect nearby robots. This would eliminate the need of communication between the robots, avoiding the negative influence that delayed communications would have in real systems, and enable the possibility to detect obstacles in the path.

Acknowledgments

The authors would like to thank the institution CAPES for providing research grant and LMI for providing the necessary resources for the development of the research.

References

[1] Merino, L., Caballero, F., De Dios, J. R. M., Maza, I., Ollero, A. Automatic forest fire monitoring and measurement using unmanned aerial vehicles, Proceedings of the 6th International Congress on Forest Fire Research: 2010.

[2] Ollero, A., Merino, L. Unmanned aerial vehicles as tools for forest-fire fighting, Forest Ecology and Management: p. 263, 2006.

[3] Choi, S. S., Kim, E. K. Building crack inspection using small UAV, 2015 17th International Conference on Advanced Communication Technology (ICACT): 235-238, 2015.

[4] Pederi, Y. A., Cheporniuk, H. S. Unmanned Aerial Vehicles and new technological methods of monitoring and crop protection in precision agriculture, 2015 IEEE International Conference In Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD): 298-301, 2015.

[5] Baker, C., Ramchurn, G., Teacy, L., Jennings, N. Planning search and rescue missions for UAV teams: 2016.

[6] Geng, L., Zhang, Y. F., Wang, P. F., Wang, J. J., Fuh, J. Y., Teo, S. H. UAV surveillance mission planning with gimbaled sensors, 11th IEEE International Conference on Control & Automation (ICCA): 320-325, 2014.

[7] Li, T., Jiang, J., Zhen, Z., Gao, C. Mission planning for multiple UAVs based on ant colony optimization and improved Dubins path, 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC): 954-959, 2016.

[8] Kushleyev, A., Mellinger, D., Powers, C., Kumar, V. Towards a swarm of agile micro quadrotors, Autonomous Robots: 287-300, 2013.

[9] Bürkle, A., Segor, F., Kollmann, M. Towards autonomous micro uav swarms, Journal of intelligent & robotic systems: 339-53, 2001.

[10] Vásárhelyi, G., Virágh, C., Somorjai, G., Tarcai, N., Szörényi, T., Nepusz, T., Vicsek, T. Outdoor flocking and formation flight with autonomous aerial robots, IEEE/RSJ International Conference on Intelligent Robots and Systems: 3866-3873, 2014.

[11] De Benedetti, M., D'Urso, F., Messina, F., Pappalardo, G., Santoro, C. UAV-based Aerial Monitoring: A Performance Evaluation of a Self-Organising Flocking Algorithm, 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing: 248-255, 2015.

[12] Gu, Y., Seanor, B., Campa, G., Napolitano, M. R., Rowe, L., Gururajan, S., Wan, S. Design and flight testing evaluation of formation control laws, IEEE Transactions on Control Systems Technology: 1105-1112, 2006.

[13] Ambroziak, L., Gosiewski, Z. Two stage switching control for autonomous formation flight of unmanned aerial vehicles, Aerospace Science and Technology 46: 221-226, 2015.

[14] Egerstedt, M., Hu, X., Stotsky, A. Control of mobile platforms using a virtual vehicle approachm, IEEE transactions on automatic control: 1777-1782, 2001.

[15] Askari, A., Mortazavi, M., Talebi, H. A. UAV formation control via the virtual structure approach, Journal of Aerospace Engineering: v. 28, n. 1, p. 04014047, 2013.

[16] Balch, T., Arkin, R. C. Behavior-based formation control for multirobot teams, IEEE transactions on robotics and automation: 926-939, 1998.

[17] Virágh, C., Vásárhelyi, G., Tarcai, N., Szörényi, T., Somorjai, G., Nepusz, T., Vicsek, T. Flocking algorithm for autonomous flying robots, Bioinspiration & biomimetics: v. 9, n. 2, p. 025012, 2014.

[18] Pixhawk homepage: https://pixhawk.org/. Acessed: September, 2016.

[19] ROS homepage: http://www.ros.org/. Acessed: September, 2016.

Distributed circular formation flight of fixed-wing aircraft with Paparazzi autopilot

Hector Garcia de Marina* Gautier Hattenberger†
ENAC, University of Toulouse, F-31055, France

ABSTRACT

In this paper we introduce the usage of guidance vector fields for the coordination and formation flight of fixed-wing aircraft. In particular, we describe in detail the technological implementation of the formation flight control for a fully distributed execution of the algorithm by employing the open-source project Paparazzi. In this context, distributed means that each aircraft executes the algorithm on board, each aircraft only needs information about its neighbors, and the implementation is straightforwardly scalable to an arbitrary number of vehicles, i.e., the needed resources such as memory or computational power not necessarily scale with the number of total aircraft. The coordination is based on commanding the aircraft to track circumferences with different radii but sharing the same center. Consequently, the vehicles will travel different distances but with the same speeds in order to control their relative angles in the circumference, i.e., their orbital velocities. We show the effectiveness of the proposed design with actual formation flights during the drone parade in IMAV2017.

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have emerged as powerful platforms for, among others, atmospheric research, intensive agriculture and surveillance. A key aspect of the usage of these vehicles is to make affordable and accessible certain tasks within these fields. In particular, the technologies around these aerial vehicles, such as batteries, propulsion systems, construction materials and their associated maintenance, have developed an interesting performance for the actual cost. This fact makes interesting the usage of several of these vehicles in a cooperative way in order to enhance tasks that were previously performed by a single aircraft. For example, we can perform missions more efficiently by splitting the payload of one aircraft, such as antennas [1] or different kind of surveillance cameras [2] to two or more aircraft. Furthermore, we can also employ the coordination of several aircraft for the estimation of the wind without employing expensive sensors [3]. Another interesting and different usage of fleets of UAVs are related with performance shows such as the suggested outdoor parade contest of IMAV2017[1].

There exist a large number of algorithms in literature related with the formation control of vehicles [4]. However, most of them consider the dynamics of the vehicle as kinematic points, which is a very restrictive requirement for fixed-wing aircraft that can be seen as an unicycle. Formation control algorithms dealing with this kind of dynamics can still be found in the literature [5, 6, 7]. However, most of them do not consider desirable requirements such as vehicles with constant speed, e.g., with motors operating constantly at their nominal conditions, or it is not clear how to combine such algorithms with trajectory tracking, which is quite important in order to guarantee the confinement of the UAVs in their allowed airspace.

This paper aims to show how the algorithm proposed in [8] has been implemented in a fully distributed way by employing the open-source project Paparazzi. The algorithm proposes a technique based on guidance vector fields for trajectory tracking [9] in order to achieve the coordination of multiple fixed-wing aircraft flying at constant (and equal) speeds. In particular, we focus on circular trajectories but it is also applicable to any other closed trajectories. The main idea relies on controlling the traveled distance by the aircraft by commanding them to track a smaller or bigger radius with respect to the desired circle. Consequently, the vehicles will travel different distances but with the same speeds in order to control their relative angles in the circumference, i.e., the algorithm controls their orbital velocities. The aircraft are not always placed on the commanded circles but far away from them, specially in the beginning of the algorithm where the initial positions of the vehicles could be arbitrary. Nevertheless, the proposed strategy is feasible because of the exponential convergence property to the desired trajectory of the guidance algorithm [9]. Therefore if the convergence of the formation flight algorithm is slow enough, then the whole cascaded system (trajectory tracking and formation control) will converge to the desired coordinated formation flight.

We organize this paper as follows. The Section 2 reviews the concept of vector field for tracking smooth trajectories. Then, we explain in Section 3 how to manipulate such vector fields in order to achieve the desired circular formation flight. We continue in Section 4 showing how the algorithm is fully distributed and executed in the open-source Paparazzi

*hector.garcia-de-marina@enac.fr
†gautier.hattenberger@enac.fr

[1] http://www.imavs.org/2017

autopilot [10]. In particular, in Section 5 we show as an example a simulation of the formation flight parade that will be performed during IMAV 2017. We end the paper with some conclusions in Section 6.

2 GUIDANCE VECTOR FIELD FOR TRACKING SMOOTH TRAJECTORIES

2.1 Fixed-wing aircraft's model

Consider for the *unit speed* fixed-wing aircraft the following nonholonomic model in 2D

$$\begin{cases} \dot{p} &= m(\psi) \\ \dot{\psi} &= u_\psi, \end{cases} \tag{1}$$

where $p = \begin{bmatrix} p_x & p_y \end{bmatrix}^T$ is the Cartesian position of the vehicle, $m = \begin{bmatrix} \cos(\psi) & \sin(\psi) \end{bmatrix}^T$ with $\psi \in (-\pi, \pi]$ being the attitude *yaw* angle[2] and u_ψ is the control action that will make the aircraft to turn. If we consider that the altitude of the vehicle is kept constant and its pitch angle is close to zero, then the control action u_ψ corresponds to the following bank angle ϕ in order to have a coordinated turn

$$\phi = \arctan \frac{u_\psi}{g}, \tag{2}$$

where g is the gravity acceleration.

2.2 Trajectory tracking

We have chosen the algorithm proposed in [9] for the task of tracking circular trajectories since it has been successfully validated in real flights [11]. One interesting property of the chosen algorithm is that the local exponential converge to the desired path is guaranteed. This property help us to support the convergence of the *high level* formation control algorithm under the argument of *slow-fast* dynamical systems [8].

Consider the circular path $\mathcal{P}$ described by

$$\mathcal{P} := \{p \, : \varphi(p) := p_x^2 + p_y^2 - r^2 = 0\}. \tag{3}$$

Clearly the function $\varphi : \mathbb{R}^2 \to \mathbb{R}$ belongs to the C^2 space and it is *regular* everywhere excepting at the origin, i.e.,

$$\nabla\varphi(p) = \begin{bmatrix} 2p_x \\ 2p_y \end{bmatrix} \neq 0 \iff p \in \mathbb{R}^2 \setminus 0. \tag{4}$$

The trajectory tracking algorithm employs the level sets $e(p) \triangleq \varphi(p)$ for the notion of *error distance* between the aircraft and $\mathcal{P}$. In particular, for circular trajectories a positive level set corresponds to an *expanded version* of the desired circle $\mathcal{P}$, while a negative level set corresponds to a *contracted version*. Note that the domain of the error distance is $e \in [-r^2, \infty)$. We define by $n(p) := \nabla\varphi(p)$ the normal vector to the curve corresponding to the level set $\varphi(p)$ and the tangent vector τ at the same point p is given by the rotation

$$\tau(p) = En(p) = \begin{bmatrix} 2p_y \\ -2p_x \end{bmatrix}, \quad E = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Note that E will determine in which direction $\mathcal{P}$ will be tracked, which is done by following the direction at each point p given by of the vector field

$$\dot{p}_d(p) \triangleq \tau(p) - k_e e(p) n(p) = 2 \begin{bmatrix} p_y - k_e e p_x \\ p_x - k_e e p_y \end{bmatrix}, \tag{5}$$

where $k_e \in \mathbb{R}^+$ is a gain that defines how *aggressive* is the vector field. An example of the construction of the guidance vector field (5) is shown in Figure 1, and its visualization in Paparazzi for tracking an ellipsoidal trajectory is shown in Figure 2.

Let us define $\hat{x}$ as the unit vector constructed from the nonzero vector x. The vector field (5) is successfully tracked if we apply the following control action to (2) [9, 11]

$$u_\psi = -\left(E\hat{\dot{p}}_d\hat{\dot{p}}_d^T E\left((E - k_e e)H(\varphi)\dot{p} - k_e n^T \dot{p} n\right)\right)^T E\frac{\dot{p}_d}{||\dot{p}_d||^2} + k_d \hat{\dot{p}}^T E \hat{\dot{p}}_d, \tag{6}$$

where $H(\cdot)$ is the Hessian operator, i.e., from (4) we have that $H = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ and $k_d \in \mathbb{R}^+$ is a positive gain that determines how fast the vehicle converges to the guidance vector field. We can clearly identify two terms in the addition in (6). The first term is a feedforward component and makes the aircraft to stay on the guidance vector field (5) while the second term makes the vehicle to converge to the guidance vector field in case that the vehicle is not aligned with it.

3 GUIDANCE VECTOR FIELD AS A COORDINATING TOOL

Consider a team of n aircraft traveling all of them at the same speed. The main objective of this paper is to show how to make them rendezvous without actuating on their traveling velocities. The rendezvous will happen at the same time that the team is traveling over a desired circular trajectory $\mathcal{P}$. We will see that this is possible to achieve by controlling the traveled distance of the aircraft around $\mathcal{P}$.

3.1 Circular trajectory

We summarize in this subsection the formation control algorithm presented in [8]. Consider that the center of $\mathcal{P}$ is at the origin as in (3). Let us define the phase of the aircraft as $\theta = \text{atan2}(p_y, p_x)$ where

$$\text{atan2}(y, x) = \begin{cases} \arctan(\frac{y}{x}) & \text{if } x > 0, \\ \arctan(\frac{y}{x}) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\ \arctan(\frac{y}{x}) - \pi & \text{if } x < 0 \text{ and } y < 0, \\ +\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0. \end{cases} \tag{7}$$

[2] For our setup, the yaw angle and heading angle can be considered equal due to the absence of wind.

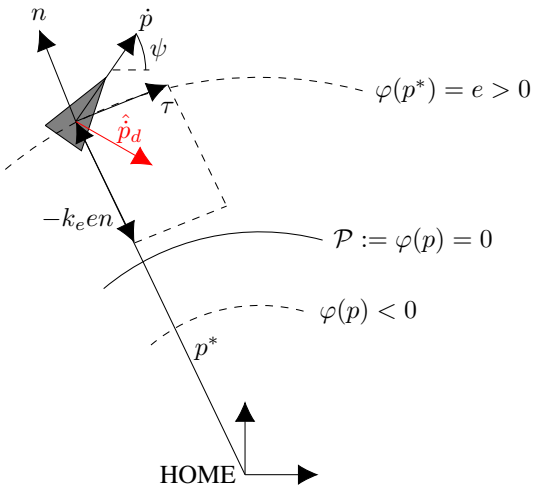


Figure 1: The direction to be followed by the UAV at the point p^* with respect to *HOME* for converging to $\mathcal{P}$ is given by $\hat{\dot{p}}_d$. The tangent and normal vectors τ and n are calculated from $\nabla\varphi(p^*)$. The error *distance* e is calculated as $\varphi(p^*)$.

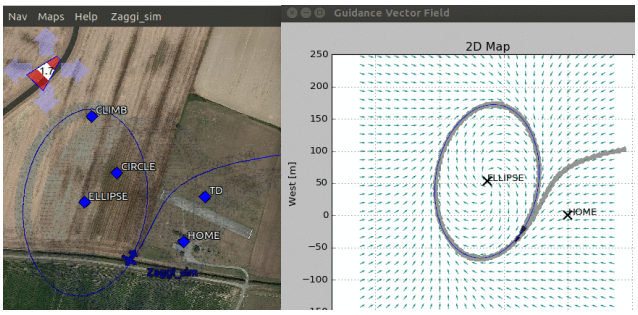


Figure 2: Example of the directions $\hat{\dot{p}}_d$ given by the vector field (5) for tracking an ellipsoidal track. Screenshot taken from the real time *gvf app* available in Paparazzi.

In a circular trajectory we are interested in controlling the different inter-vehicle phases, e.g., between aircraft 1 and 2 we could define $z_1 := \theta_1 - \theta_2$. For a general case, the relationships between neighbors are described by an undirected graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ with the vertex set $\mathcal{V} = \{1, \ldots, n\}$ and the ordered edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The set $\mathcal{N}_i$ of the neighbors of agent i is defined by $\mathcal{N}_i \triangleq \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$. We define the elements of the incidence matrix $B \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$ for $\mathbb{G}$ by

$$b_{ik} \triangleq \begin{cases} +1 & \text{if} \quad i = \mathcal{E}_k^{\text{tail}} \\ -1 & \text{if} \quad i = \mathcal{E}_k^{\text{head}} \\ 0 & \text{otherwise} \end{cases}. \tag{8}$$

Note that the i'th row of B denotes for the i'th vehicle and the k'th column denotes for the link $\mathcal{E}_k$. Define $\Theta \in \mathbb{R}^{|\mathcal{E}|}$ as the stacked vector of aircraft phases, then one can calculate

$$z = B^T \Theta, \tag{9}$$

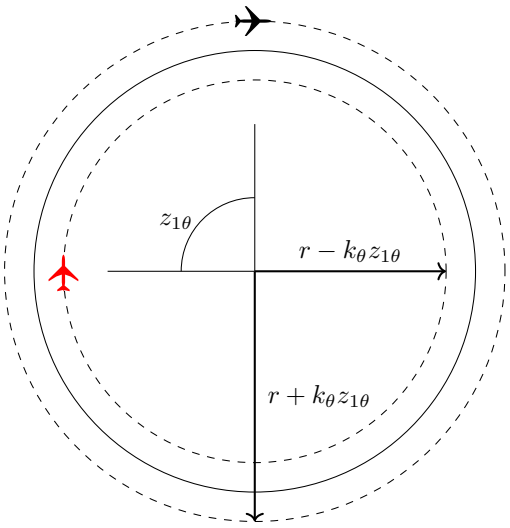


Figure 3: Red and black airplane want to fly over $\mathcal{P} := x^2 + y^2 - r^2 = 0$ (solid circle) with the same phase $\theta_1 = \theta_2$, i.e., $z_{1\theta} := \theta_1 - \theta_2 = 0$. Red airplane travels less distance if it tracks a smaller radius (negative level set with respect to $\mathcal{P}$), therefore its angular velocity with respect to the center of $\mathcal{P}$ is bigger than the angular velocity of black airplane. Once $z_{1\theta} = 0$, both aircraft are tracking $\mathcal{P}$. Setting the positive gain k_θ will tune the convergence time for the rendezvous.

as the stacked vector of all the available inter-vehicle phases $z_k(t)$ to be controlled. In particular, for rendezvous, we are interested in driving the signal $z(t)$ to zero. Depending on which level set of $\mathcal{P}$ an aircraft is following, it will travel with a different angular velocity $\dot{\theta}_i$ with respect to the center of $\mathcal{P}$, allowing us to control the evolution of the different z_k's. In order to track different level sets of $\mathcal{P}$ we introduce a control signal ${}^i u_r$ to (3) for the i'th aircraft as

$$\varphi(p) := p_x^2 + p_y^2 - (r + {}^i u_r)^2 = 0, \tag{10}$$

and we apply a consensus algorithm

$${}^i u_r = k_r B_i z, \tag{11}$$

where B_i stands for the i'th row of the incidence matrix B and $k_r \in \mathbb{R}^+$ is a positive gain. The results in [8] guarantees the exponential stability of the equilibrium at $\Theta = 0$ if $\mathcal{G}$ does not contain any cycles, e.g., a spanning tree topology. An example for two aircraft can be found in Figure 3.

4 Implementation in Paparazzi

We discuss in this section the implementation of the formation flight algorithm in the open-source project Paparazzi. We start by rewriting the control action (11) as follows

$${}^i u_r = k_r \sum_{j \in \mathcal{N}_i} (\theta_i - \theta_j), \tag{12}$$

$\mathcal{N}_i$	$\theta_{j\in\mathcal{N}_i}$	Timeout(ms)
2	π	1042
3	0.4π	426

Table 1: Example of table with neighbor's information.

therefore the i'th aircraft only needs to collect the phases θ_j from its neighbors. We do this process of collection by inter-vehicle communication, i.e., each aircraft calculates its phase θ_i and transmits it to its neighbors. Note that since we are controlling inter-vehicle phases, the aircraft can control their relative phases while following circumferences with different centers for $\mathcal{P}$, although obviously the rendezvous will not take place in such a situation.

We write the software responsible of running the algorithm in Paparazzi as a module[3]. A module allows one to add new code in a flexible way with initialization, periodic and event functions without modifying the main autopilot loop. Each aircraft will run exactly the some code without any particular modification. This will allow the scaling up of the number of aircraft without any associated penalty. In order to run the algorithm each aircraft is required in their flight plan to be tracking a circle with the guidance vector field algorithm[4].

The module employs the new functionalities of Paparazzi Link v2.0[5]. This communication layer allows the inter-vehicle communication, i.e., air to air, without intervention of the Ground Control Station. Each aircraft in Paparazzi has an unique identification ID (uint8) that will be employed for each aircraft i in (12). The formation flight module runs two independent processes. The first process keeps updated a table once a message from a neighbor is received. This table shown in Table 1 has information about the IDs of the neighbors, their latest received θ_j and the time since this value was updated. An aircraft can always ask to be registered in or deleted from another aircraft's table in order to become neighbors or break the relationship. The second process is executed periodically with a frequency of 2Hz. It calculates ${}^iu_r$ in (12) from all the updated data not older than 2 seconds, so we avoid situations like an aircraft that abandoned the formation but it continues registered in its neighbors table. Then the aircraft updates the radius to be tracked by the guidance vector field. Finally, the aircraft updates its own θ_i and transmit it to its neighbors if and only if the GPS is reliable, e.g., it has *3D Fix*. Consequently, if the aircraft does not update its neighbors' tables, then it will not be taken into account in the formation after the timeout. This process is summarized in Algorithm 1.

[3]https://wiki.paparazziuav.org/wiki/Modules

[4]https://wiki.paparazziuav.org/wiki/Module/guidance_vector_field

[5]Paparazzi Link is a messages toolkit (message definition, code generators, libraries) to be used with Paparazzi and compatible systems

Result: Rendezvous of aircraft i with its neighbors.
Data: θ_i and Table 1.
while *Formation Control == True* **do**
  ${}^iu_r = 0$;
  for *All the rows of the table* **do**
    if *Timeout is not reached* **then**
      ${}^iu_r = {}^iu_r + (\theta_i - \theta_j)$;
    end
  end
  Set radius $r^2 + {}^iu_r$ in the guidance vector field;
  if *GPS is reliable* **then**
    Transmit θ_i to $\mathcal{N}_i$;
  end
end

Algorithm 1: Algorithm executed at aircraft i once the formation control is activated. Note that is not only distributed but it does not need all the information from all the neighbors to be executed.

5 IMAV FORMATION FLIGHT PARADE

The three employed aircraft are custom 1.5m wingspan flying-wings equipped with the following electronics:

- 2x servos SAVOX SH-0257MG
- ESC Flyduino KISS 18A v1.2
- Brushless motor T-Motor MT2208-18 1100KV
- Propeller 8x6
- 3S Battery pack built from Panasonic NCR18650B (up to 40min endurance, 2h+ with 2x3S)
- Apogee board autopilot[6]
- Futaba receiver R6303SB
- Futaba transmitter FAAST
- 2x XBEE Pro S1 (on board and on ground)
- GPS M8
- Ground Control Station: Laptop running Paparazzi on Ubuntu 17.04

We perform a formation flight simulation of three aircraft planned for the parade show in IMAV2017. For real experiment results like the one from the Figure 5 we refer to [8]. Although the available space for maneuvering is tight, the simulation shows that it is possible to synchronize the aircraft flying at 11m/s (ground speed) in circumferences of radius 30 meters as it is shown in Figure 6.

[6]https://wiki.paparazziuav.org/wiki/Apogee/v1.00

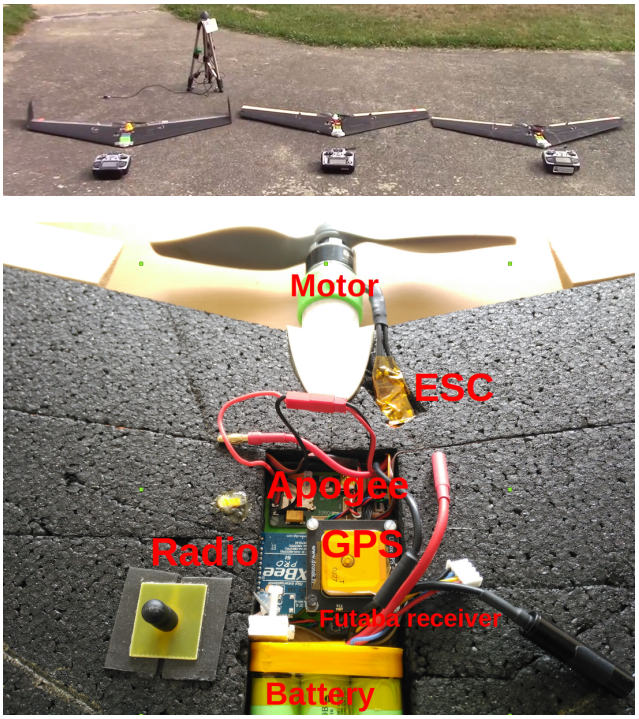


Figure 4: The three custom flying-wings employed for the formation flight parade equipped with Paparazzi autopilot.



Figure 5: Picture taken from an actual formation flight executing the algorithm described in this paper.

6 Conclusions

In this paper we have presented a fully distributed implementation of a flight formation controller for fixed-wing aircraft in the open-source autopilot Paparazzi. We manipulate guidance vector fields in order to control the inter-vehicle positions by changing the radius of the circumference to be eventually tracked. We finally show the effectiveness of the proposed strategies based on the rules for the IMAV2017 drone parade outdoor competition.

References

[1] Shuanggen Jin and Attila Komjathy. Gnss reflectometry and remote sensing: New objectives and results. *Advances in Space Research*, 46(2):111–117, 2010.

[2] Iván Maza, Fernando Caballero, Jesús Capitán, José Ramiro Martínez-de Dios, and Aníbal Ollero. Experimental results in multi-uav coordination for disaster management and civil security applications. *Journal of intelligent & robotic systems*, 61(1):563–585, 2011.

[3] Stephanie Mayer, Gautier Hattenberger, Pascal Brisset, Marius O Jonassen, and Joachim Reuder. A no-flow-sensorwind estimation algorithm for unmanned aerial systems. *International Journal of Micro Air Vehicles*, 4(1):15–29, 2012.

[4] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

[5] Joshua A Marshall, Mireille E Broucke, and Bruce A Francis. Pursuit formations of unicycles. *Automatica*, 42(1):3–12, 2006.

[6] Rodolphe Sepulchre, Derek A Paley, and Naomi Ehrich Leonard. Stabilization of planar collective motion with limited communication. 53(3):706–719, 2008.

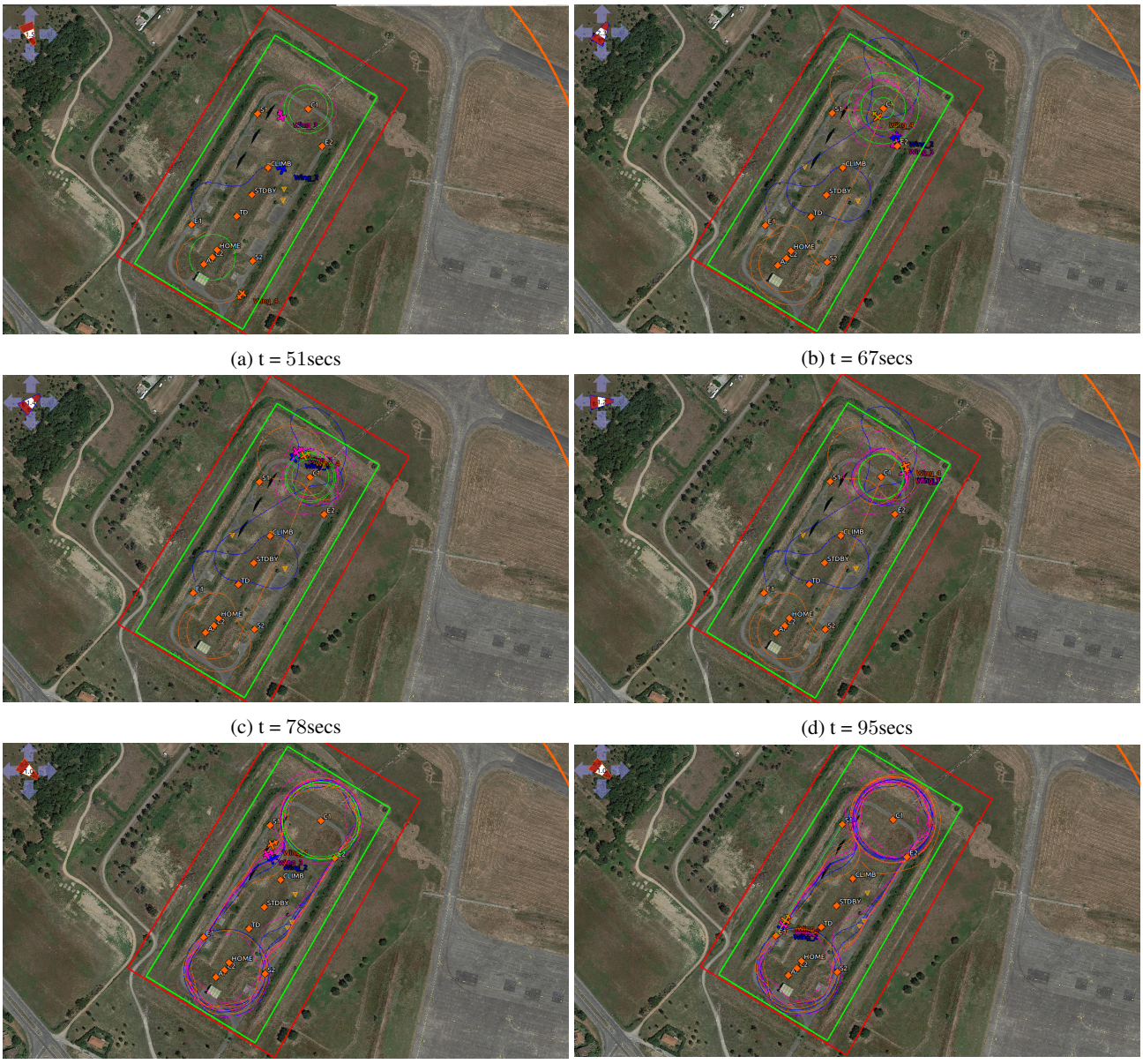
[7] Z. Sun, H. G. de Marina, G. S. Seyboth, B. D. O. Anderson, and C. Yu. Circular formation control of multiple unicycle-type agents with non-identical constant speeds. *IEEE Transactions on Control Systems Technology*, 2017.

[8] Hector G. de Marina, Zhiyong Sun, Murat Bronz, and Gautier Hattenberger. Circular formation control of fixed-wing uavs with constant speeds. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, 2017.

[9] Yuri A. Kapitanyuk, Anton V. Proskurnikov, and Ming Cao. A guiding vector field algorithm for path following control of nonholonomic mobile robots. *IEEE Transactions on Control System Technology*, 2016.

[10] Gautier Hattenberger, Murat Bronz, and Michel Gorraz. Using the paparazzi UAV system for scientific research. In *IMAV 2014, International Micro Air Vehicle Conference and Competition 2014*, pages pp–247, 2014.

[11] Hector Garcia de Marina, Yuri A Kapitanyuk, Murat Bronz, Gautier Hattenberger, and Ming Cao. Guidance algorithm for smooth trajectory tracking of a fixed wing uav flying in wind flows. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.



(a) t = 51secs (b) t = 67secs

(c) t = 78secs (d) t = 95secs

(e) (f)

Figure 6: Screenshots from the Paparazzi ground control station showing the evolution of the circular formation. The first screenshot starts with the aircraft at arbitrary positions within the allowed flying area for the IMAV outdoor competition. The convergence to a synchronized formation flight takes place in around thirty seconds without leaving the allowed flying area. The circular synchronization can be used for *launching* the aircraft to travel together over the same segment.

EDURA: an Evolvable Demonstrator for Upset Recovery Approaches with a 3D-printed Launcher

Torbjørn Cunis* and Murat Bronz†
French Civil Aviation School, 7 avenue Edouard Belin, Toulouse, France

Abstract

As in-flight loss of control has remained a severe threat to aviation, aeronautical research designed several approaches for upset recovery, few of which has been demonstrated in flight tests. The on-going success of micro air vehicles, however, rises the possibility of cheap and flexible flight demonstrations. In this paper, we present the concept of and first steps towards an aerial experimental platform for upset recovery: EDURA.

EDURA is part of the CONVEX project to investigate, develop, and demonstrate non-linear upset recovery control laws in a fixed-wing MAV.

1 Introduction

In-flight loss of control (LOC-I) imposes the *highest risk to aviation safety* [1] and has remained the foremost cause of fatal accidents for the last decades. Generally defined as any *deviation from the desired flight-path* by [2], LOC-I especially includes upset situations such as stall, high and inverted bank angle, as well as post-stall spirals and rotations [3]. With their unstable and highly non-linear characterizations, these situations require extensive control effort and adequate approaches to recover the upset aircraft and return into the flight envelope.

Non-linear behaviour of aircrafts in the post-stall flight regime has been investigated analytically [4–8] and in wind-tunnel studies [9, 10]. As a result, researchers developed control laws for upset recovery [11–20]. For the recovery approaches found in literature as well as proposed by the authors in [21] are model-based, there is a need for reliable flight dynamics data. However, though the NASA *generic transport model* (GTM) offers a scaled unmanned aerial platform, well-investigated in wind-tunnel studies, to test control systems [22], only Gregory et al. [20] report flight tests of the designed upset recovery approach.

In the past decade, the market for micro air vehicles (MAVs) has grown considerably. Widely available now, MAVs both offer cheap and repeatable experiments while being easy to replace and maintain in case of unsuccessful tests. On the other hand, indoor flight tests provide several benefits such as availability of accurate position tracking systems and reduction of disturbances in the test area, while requiring a small-scale vehicle other than the GTM.

Combining the mentioned points—accurate aerodynamic data, indoor tests, and usage of established MAV supply chains—, in this paper we are going to present concept and first version of a small-scale, fixed-wing experimental platform resembling an easy-to-model flying plate; based on a commercial-off-the-shell aircraft, for the first version, and the open-source autopilot software *Paparazzi UAV*. Furthermore, we will propose and present a catapult launcher allowing repeatable and configurable test conditions including those which actually pose an upset, like insufficient air speed, high angle of attack, or inverted bank angle. Based on a CAD model of the aircraft, we conclude with deriving its aerodynamic coefficients.

2 EDURA Concept

Indoor flight arenas, as existing at several research sites today, provide an ideal test environment for unmanned air vehicles. Wind and gust disturbances are reduced due to their closed walls and optical tracking systems provide position, attitude, and velocity information at both high accuracy and frequency. However, they are limited in space and hence unsuitable for larger vehicles. While clearly benefiting from indoor flight conditions, a suitable fixed-wing MAV is mainly required to be small.

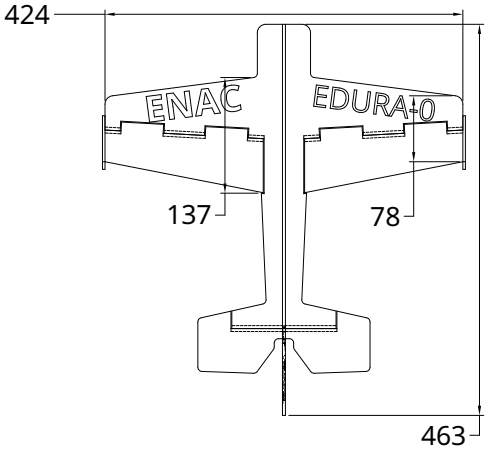
Depron is a light-weight material which allows effortless processing. Wings cut of a single layer of depron show a rectangular surface and thus can be modelled as flying plate to obtain the aerodynamics coefficients. An aircraft made of depron offers design, making, and aerodynamic modelling of an aerial experimental platform in short iterations, evolving the flight dynamics as suitable. Furthermore, it accounts for the small-scale design necessary for indoor flight tests. Today's miniature microcontrollers, sensors and actuators, and auxiliary boards complete the setup albeit deliver full flight control and navigation on-board.

Eliminating the necessity of propulsion, a catapult launcher initially accelerates the aircraft to the desired air speed. In addition, it provides a configurable initial angle of attack and flight-path angle. The configuration of all three air speed, angle of attack, and flight-path angle after launch is repeatable over multiple executions of a single test case.

*Doctoral researcher at ENAC and ONERA – The French Aerospace Lab; torbjoern.cunis@recherche.enac.fr
†Assistant professor at ENAC; murat.bronz@enac.fr



(a) Side view.



(b) Top view.

Figure 1: Side view (a) of the EDURA-0 vehicle based on an *E-flite UMX Yak 54 3D* commercial-off-the-shell aircraft; and CAD drawing (b) from the top with measures: wing span, root chord, tip chord, and length (Quantities in millimetres).

3 EDURA-0 Aircraft

The first aircraft is based on an *E-flite UMX Yak 54 3D* commercial off-the-shelf radio-controlled aircraft,[1] as shown in Fig. 1a. This 35 g vehicle with a wing span of 42.4 cm and length of 46.3 cm is made of 35 mm thick depron and resembles a flying plate with trapeze-shaped wings (Fig. 1b). The control surfaces, elevator, rudder, and left and right ailerons are driven by four linear servos at a speed of 0.14 s [23].

The E-flite aircraft has been originally shipped with a *Spectrum* radio receiver, which was replaced by a *Lisa-MXs* autopilot board [24] and an *ESP8266-9* wifi module for UDP-based communication (Fig. 2). The autopilot runs the open-source software *Paparazzi UAV*.[2]

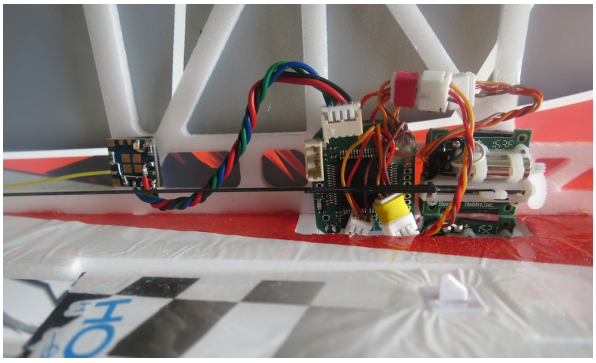


Figure 2: Autopilot configuration (from left to right): ESP8266-9 wifi module, Lisa-MXs board, and the linear servo for the elevator.

For unpropelled flight, the propellor and its motor will be removed and replaced by a respective weight for balance.

[1] http://www.e-fliterc.com/Products/?ProdID=EFLU3550
[2] http://paparazziuav.org

4 Catapult Launcher

There are three main requirements to a catapult launcher for upset recovery tests, as aforementioned: to accelerate the aircraft to a certain air speed; to establish the initial angle of attack; and define the flight-path, *i.e.* the flight-path angle and heading.

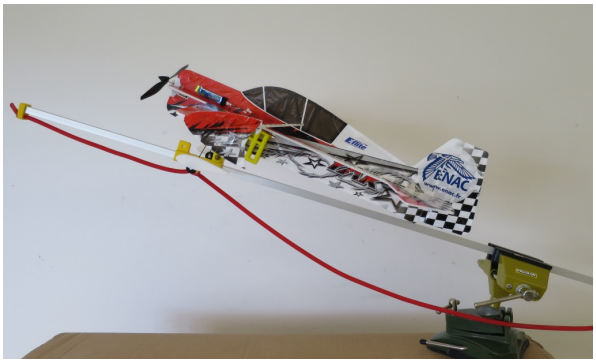
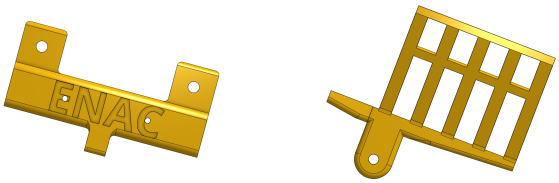


Figure 3: The EDURA catapult launcher with aircraft.

As for those, the catapult launcher is designed of three components (Figs. 3 and 4), the rail, a cart moving lateral along the rail, and a cage to carry the aircraft. The cart is accelerated by an elastic band fixed to the front of the rail, while the same band attached to the end, too, slows down the cart after ejecting the aircraft.

Rail The rail of the catapult launcher is based on a 1 m aluminium tube with quadratic surface and inner and outer edges of 8 mm and 10 mm, respectively. Front and back end stop-



(a) Cart. (b) Cage.

Figure 4: Components of the catapult launcher in CAD.

pers were 3D-printed to attach the elastic band to as well as to prevent the cart from racing beyond the rail's ends. The orientation of the rail with respect to the global reference frame will define the flight-path vector of the aircraft.

Cart The cart has been designed in CAD (Appendix, Fig. 6) and was 3D-printed afterwards. Its quadratic body fits around the rail and allows lateral motion of one degree of freedom. The legs at the back of the cart allows the cage to be attached with a variable angle, defining the angle of attack. At the bottom, an eye is provided for the elastic band.

Cage Just as the cart, the cage was 3D-printed. Attached on top of the cart, it holds the aircraft during the acceleration and allows a smooth ejection. A CAD drawing of the cage is shown in the appendix (Fig. 7).

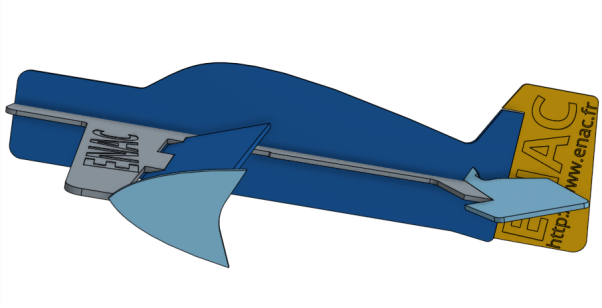
The initial air-speed, *i.e.* the speed of the cart with respect to the rail can be configured by pulling the cart backwards thus stretching the elastic to a certain length with respect to its resting point.

5 Aerodynamic Coefficients

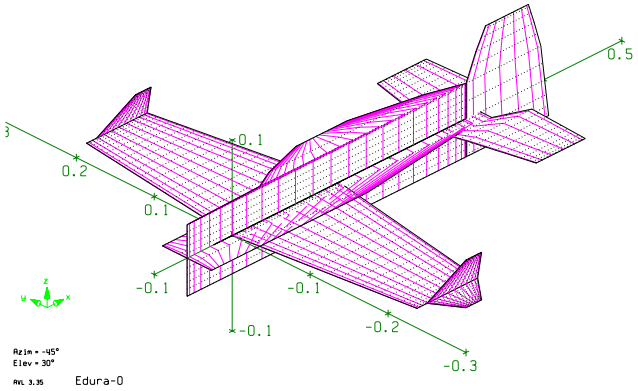
Fuselage, wings, and control surfaces of the UMX Yak 54 3D aircraft were measured and modelled in CAD (Fig. 5a). Aerodynamic coefficients are obtained numerically by using a program based on vortex-lattice method, called AVL[3] (Fig. 5b). As a result, the linearized stability derivatives around the selected operating point are given in Tab. 1 for 4 m/s cruise speed and 15 % static margin. Based on the initial numerical estimation of the coefficients, a comparison can be made between the expected and measured flight trajectories. According to the comparison, it will be possible to verify the linear coefficients and identify the non-linear part in order to extend and improve the numerical estimation of the coefficients. An improved model, however, is crucial for the development of upset recovery approaches.

6 Conclusion

In this paper, we have argued the need of an experimental platform for flight tests of upset recovery approaches. We have therefore discussed the benefits of indoor flights of suitable MAVs and proposed the EDURA concept of an evolvable



(a) CAD model.



(b) AVL mesh.

Figure 5: CAD model of the EDURA-0 aircraft and its control surfaces (a) and AVL Mesh (b).

small-scale, fixed-wing MAV as demonstrator. An additional catapult launcher allows repetition of the initial flight conditions. The first prototype based on a commercial-off-the-shelf aircraft has as well been presented as the design and development of the catapult launcher.

The system illustrated is to be evaluated in launch tests with indoor position tracking in order to proof repeatability of the initial conditions. By free-flight force estimation and force measurement in wind-tunnel tests, future studies would verify the aerodynamic coefficients derived on the aircraft model. Overall, we have introduced an aerial experimental platform to test and demonstrate upset recovery approaches based on the aerodynamic model of the aircraft, where the *modus operandi* is going to allow the evolution of the vehicle's parameters and test conditions.

Acknowledgements

The authors wish to thank the TU Delft Micro Air Vehicle Laboratory for providing the Lisa-MXs hardware and support.

CAD models of aircraft and aircraft launcher were created in the free-to-use cloud platform *Onshape*.[4]

[3] http://raphael.mit.edu/avl

[4] http://onshape.com

	α	β	p	q	r	δ_{elv}	δ_{ail}	δ_{rud}
C_L	4.4478	0.0	0.0	8.2844	0.0	0.01506	0.0	0.0
C_Y	0.0	-0.7016	0.25792	0.0	0.82037	0.0	0.000254	0.0081
$C_{D_{ff}}$	-	-	-	-	-	0.001086	0.0	0.0
e	-	-	-	-	-	0.000486	0.0	0.0
C_l	0.0	-0.07259	-0.4150	0.0	0.09355	0.0	-0.008842.	0.0002
C_m	-0.6156	0.0	0.0	-10.133	0.0	-0.03268	0.0	0.0
C_n	0.0	0.14285	-0.09723	0.0	-0.44895	0.0	-0.000924	-0.00538

Table 1: Stability derivatives extracted from AVL program for the aircraft at 5 m/s equilibrium cruise speed. All derivatives are in 1/rad or s/rad except for the control derivatives δ, which are in $1/^\circ$.

References

[1] Unstable Approaches: Risk Mitigation Policies, Procedures and Best Practices. Technical report, International Air Transport Association, Montreal, CA, 2015.

[2] Airplane Flying Handbook. FAA handbook FAA-H-8083-3B, Flight Standards Service, Washington, US-DC, 2016.

[3] J. R. Chambers and S. B. Grafton. Aerodynamic Characteristics of Airplanes at High Angles of Attack. NASA technical memorandum NASA/TM-74097, Langley Research Center, Hampton, US-VA, 1977.

[4] James V Caroll and Raman K Mehra. Bifurcation Analysis of Nonlinear Aircraft Dynamics. *Journal of Guidance, Control, and Dynamics*, 5(5):529–536, 1982.

[5] Craig C Jahnke. *Application of Dynamical Systems Theory to Nonlinear Aircraft Dynamics*. Phd thesis, California Institute of Technology, Pasadena, US-CA, 1990.

[6] M.G. Goman, G.I. Zagainov, and A.V. Khramtsovsky. Application of bifurcation methods to nonlinear flight dynamics problems. *Progress in Aerospace Sciences*, 33(9–10):539–586, 1997.

[7] Harry G Kwatny, Jean-Etienne T Dongmo, Bor-chin Chang, Gaurav Bajpai, Murat Yasar, and Christine Belcastro. Nonlinear Analysis of Aircraft Loss of Control. *Journal of Guidance, Control, and Dynamics*, 36(1): 149–162, 2013.

[8] Jacobus Adriaan Albertus Engelbrecht, Simon J. Pauck, and Iain K. Peddle. A Multi-mode Upset Recovery Flight Control System for Large Transport Aircraft. In *AIAA Guidance, Navigation, and Control Conference*, Boston, US-MA, aug 2013.

[9] John V Foster, Kevin Cunningham, Charles M Fremaux, Gautam H Shah, and Eric C Stewart. Dynamics Modeling and Simulation of Large Transport Airplanes in Upset Conditions. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, US-CA, aug 2005.

[10] Neal T Frink, Patrick C Murphy, Harold L Atkins, Sally A Viken, Justin L Petrilli, Ashok Gopalarathnam, and Ryan C Paul. Computational Aerodynamic Modeling Tools for Aircraft Loss of Control. *Journal of Guidance, Control, and Dynamics*, 0(0), 2016.

[11] Frank W Burcham Jr, John J Burken, Trindel A Maine, and C Gordon Fullerton. Development and Flight Test of an Emergency Flight Control System Using Only Engine Thrust on an MD-11 Transport Airplane. NASA technical publication NASA/TP-97-206217, Dryden Flight Research Center, Edwards, US-CA, 1997.

[12] Frank W Burcham Jr, Richard Stevens, Ronald Broderick, and Kerry Wilson. Manual Throttles-Only Control Effectiveness for Emergency Flight Control of Transport Aircraft. In *9th AIAA Aviation Technology, Integration, and Operations Conference*, Hilton Head Island, US-SC, sep 2009.

[13] James M Urnes Sr. Flight Control for Multi-engine UAV Aircraft using Propulsion Control. In *AIAA Infotech@Aerospace*, Garden Grove, US-CA, 2012.

[14] Bor-Chin Chang, Harry G. Kwatny, Elie R. Ballouz, and David C. Hartmann. Aircraft Trim Recovery from Highly Nonlinear Upset Conditions. In *AIAA Guidance, Navigation, and Control Conference*, San Diego, US-CA, 2016.

[15] Enric Xargay, Naira Hovakimyan, and Chengyu Cao. $\mathcal{L}_1$ adaptive controller for multi-input multi-output systems in the presence of nonlinear unmatched uncertainties. In *IEEE American Control Conference*, pages 874–879, Baltimore, US-MD, 2010.

[16] Vahram Stepanyan, Kalmanje Krishnakumar, John Kaneshige, and Diana Acosta. Stall Recovery Guidance Algorithms Based on Constrained Control Approaches. In *AIAA Guidance, Navigation, and Control Conference*, San Diego, US-CA, 2016.

[17] Jacobus Adriaan Albertus Engelbrecht. *Automatic Flight Envelope Recovery for Large Transport Aircraft*.

Phd thesis, University of Stellenbosch, Matieland, ZA, 2016.

[18] Luis G Crespo, Sean P Kenny, David E Cox, and Daniel G Murri. Analysis of Control Strategies for Aircraft Flight Upset Recovery. In *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, US-MN, aug 2012.

[19] Nathan D Richards, Neha Gandhi, Alec J Bateman, David H Klyde, and Amanda K Lampton. Vehicle Upset Detection and Recovery for Onboard Guidance and Control. *Journal of Guidance, Control, and Dynamics*, 0(0), 2016.

[20] Irene Gregory, Enric Xargay, Chengyu Cao, and Naira Hovakimyan. Flight Test of $\mathcal{L}_1$ Adaptive Control Law: Offset Landings and Large Flight Envelope Modeling Work. In *AIAA Guidance, Navigation, and Control Conference*, Portland, US-OR, aug 2011.

[21] Torbjørn Cunis, Laurent Burlion, and Jean-Philippe Condomines. Non-linear Analysis and Control Proposal for In-flight Loss-of-control. In *20th IFAC World Congress*, Toulouse, FR, 2017.

[22] Thomas L Jordan, John V Foster, Roger M Bailey, and Christine M Belcastro. AirSTAR: A UAV Platform for Flight Dynamics and Control System Testing. In *AIAA Aerodynamics Measurement Technology and Ground Testing Conference*, San Francisco, US-CA, 2006.

[23] Spektrum AR6410/AR6410L User Guide. Technical report, Horizon Hobby, Inc., 2012.

[24] Kimberly McGuire, Guido de Croon, Christophe De Wagter, Karl Tuyls, and Hilbert Kappen. Efficient Optical flow and Stereo Vision for Velocity Estimation and Obstacle Avoidance on an Autonomous Pocket Drone. *IEEE Robotics and Automation Letters*, 2(2): 1070–1076, 2017.

Appendix A: CAD Drawings

CAD drawings of the cart and cage components of the catapult launcher are shown in Figs. 6 and 7, respectively.

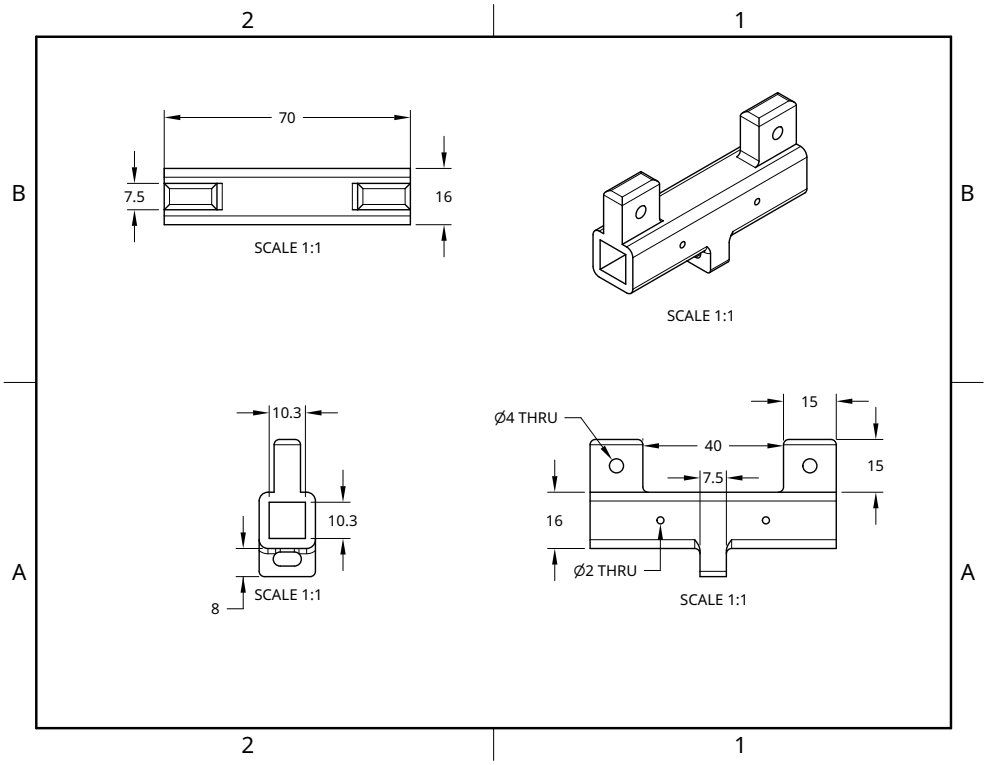


Figure 6: CAD drawing of the cart component of the catapult launcher. All quantities are given in millimetres.

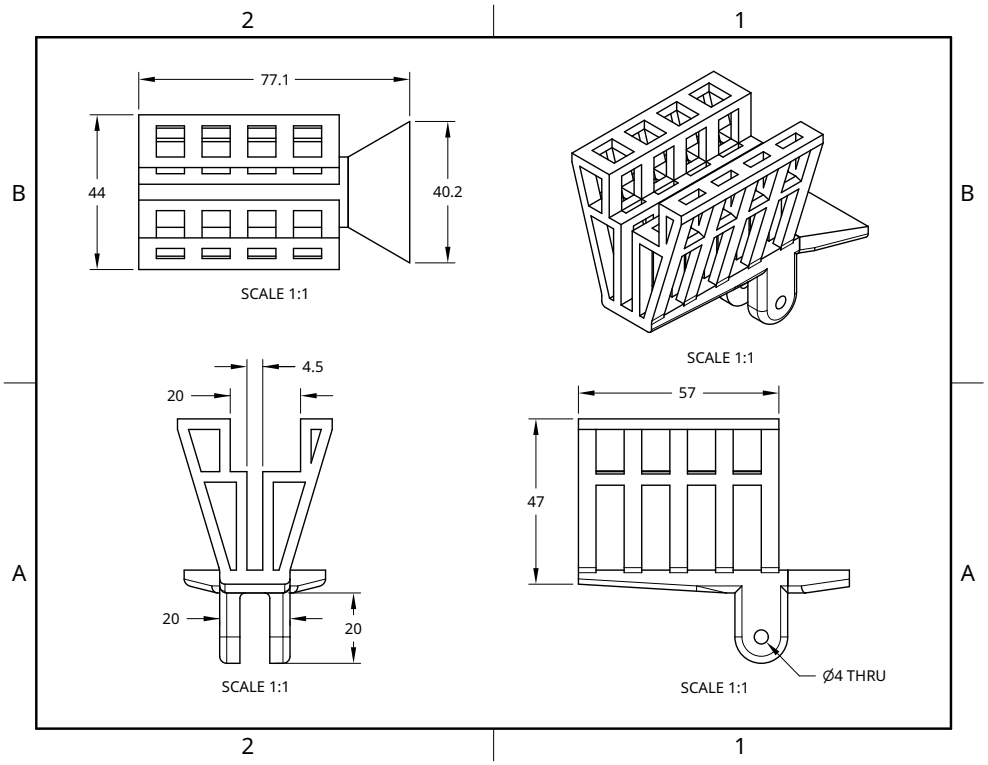


Figure 7: CAD drawing of the cage component of the catapult launcher. All quantities are given in millimetres.

Optimization of Energy Consumption for Quadrotor UAV

F. Yacef*, N. Rizoug, O. Bouhali, and M. Hamerlain
ESTACA, Laval, France
Jijel University, Jijel, Algeria
CDTA, Algiers, Algeria

ABSTRACT

In this paper we deals with the limitation of flight endurance for quadrotor unmanned aerial vehicles. Quadrotor UAVs are multi-rotors flying machines; thus, a large proportion of their energy is consumed by rotors in order to maintain the vehicle in the air. In this concept, we introduced an energetic model composed of quadrotor movement dynamic, motors dynamic and battery dynamic; then, the proposed model was validated through simulation to show possibility of saving energy. An optimal control problem is formulated and solved in order to compute minimum energy. In this problem, we seeks to find control inputs and vehicle trajectory between initial and final configurations that minimize the consumed energy during a specific mission. Simulation experiment is made for a quadrotor to highlight the proposed optimization method.

1 INTRODUCTION

In the last few years, rotary wings unmanned aerial vehicles UAVs have attract more interest due to the wide range of applications that can be addressed with such a vehicle. Recently some promising new applications have been emerged like package delivery, cinematography, agriculture surveillance and aerial manipulation; however theses applications still restricted since the embedded energy whose source is LiPo battery provide a flight time between 15 and 45 minutes, which limited the class of mission that can be carried out successfully by the UAV. To undertake this problem, many efforts have made in reduction of rotary wings UAVs weight by the use of carbon fiber airframe and high energy density intelligent-soft and in the improvement of power to weight rate. These solutions success to reduced operating in energy-starved regimes; nevertheless, no significant technological progress is made until now.

Many studies have been proposed recently contributing towards save energy and increased endurance. These contributions have mainly focused on the design of automated battery charging/replacing system. A battery swapping system for multiple small-scale UAVs have been proposed in [1]. In addition to the battery swapping mechanism, the system includes an online algorithm that can supervise replenishment of many UAVs operating simultaneously, determine when the vehicle require replenishment and perform a precision landing onto the battery swapping mechanism's landing platform. In [2] the design, test and construction of an autonomous ground recharge station for battery-powered quadrotor helicopter was presented. An energy management algorithm was implemented for a multi-agent system where the priority is given to the group to ensure the optimality of the solution regardless of number, position and density of the environment. Where in [3], an autonomous battery maintenance mechatronic system to extends the operational time of battery powered small-scaled UAVs have been developed.

Other studies have introduce endurance estimation model, in [4] a simple model is proposed to estimate the endurance of an indoor hovering quadrotor, whereas in [5] a characterization of the power consumption of rotorcraft supplied by LiPo battery and an accurate endurance estimation model have been introduced. In order to extends UAVs operating time a battery state of charge based altitude controller for an six-rotor aircraft was proposed in [6], where a battery monitoring system was designed in order to estimate state of charge (SOC) and then use it to calculate the designed controller.

In view of path following control with minimum energy consumption, the authors in [7] evaluate the relationship between navigation speed and energy consumption in a miniature quadrotor helicopter, which travels over a desired path through experimental test. Then, a novel path-following controller is proposed in which the speed of the rotorcraft is a dynamic profile that varies with the geometric requirements of the desired path.

The energy optimal path planning problem for rotary wings UAVs has gain less interest in the unmanned aerial systems literature. In [8] an approach has been proposed to solve near-minimum-energy tours for an hexarotor on a multi-target mission using the generalized traveling salesman problem with Neighborhoods and a heuristic algorithm with 4-DOF dynamic model for cost function calculation. Where in [9] minimum-energy paths were obtained between given initial and final configurations for a 6-DOF quadrotor UAV by solving an optimal control problem, also a minimum-time and/or minimum-control-effort trajectory was calculated by

*Email : fyacef@cdta.dz, fouad.yacef@estaca.fr

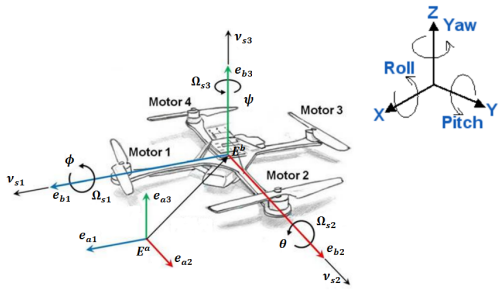


Figure 1: Quadrotor scheme

solving a related optimal control problem.

Motivated by the previous discussion, in this paper we introduced an energetic model for quadrotor UAVs. The proposed energetic model contains the vehicle dynamic, actuator dynamic and battery dynamic with an efficiency function in order to modeled motor efficiency and have an energetic model close to reality. Then, we proposed an energy optimal control problem, where the objective is to minimize the energy consumed by the quadrotor vehicle at the end of the mission while the quadrotor aircraft has to satisfy boundary conditions and feasibility constraints on the states of the system and control inputs.

The rest of this paper is organized as follows. In section 2, we presented Energetic model. Then, in section 3 Energy optimization problem is introduced. In section 4, nonlinear programming method and simulation results are presented. Conclusions and remarks are drawn in section 5.

2 Energetic model

Before proceeded in energy optimization we need an energetic model in order to have an idea about the energy consumed by the vehicle during the mission and provide a direct relation between energy and vehicle dynamic's.

2.1 Quadrotor dynamic model

In this section, we will first introduce the quadrotor unmanned aerial flying vehicles coordinate system as depicted in Fig. 1. To study the system motion dynamics, two frames are used: an inertial frame attached to the earth defined by $E^a(e_{a1}, e_{a2}, e_{a3})$ and a body-fixed frame $E^b(e_{b1}, e_{b2}, e_{b3})$ fixed to the center of mass of the quadrotor. The absolute position of the quadrotor is described by $p = [x, y, z]^T$ and its attitude by the Euler angles $\eta = [\phi, \theta, \psi]^T$. The attitude angles are respectively Yaw angle (ψ rotation around z-axis), Pitch angle ((θ rotation around y-axis), and roll angle (ϕ rotation around x-axis) [10]. The dynamic model for quadrotor vehicle can be derived as

$$\begin{aligned} m\ddot{x} &= (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)T \\ m\ddot{y} &= (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)T \\ m\ddot{z} &= (\cos\phi\cos\theta)T - mg \\ I_x\ddot{\phi} &= (I_y - I_z)\dot{\theta}\dot{\psi} + J\dot{\theta}\varpi + lu_1 \\ I_y\ddot{\theta} &= (I_z - I_x)\dot{\phi}\dot{\psi} - J\dot{\phi}\varpi + lu_2 \\ I_z\ddot{\psi} &= (I_x - I_y)\dot{\phi}\dot{\theta} + u_3 \end{aligned} \tag{1}$$

where $\varpi = \omega_1 - \omega_2 + \omega_3 - \omega_4$. where J is the rotor inertia, m, I_x, I_y and I_z denotes the mass of the quadrotor flying vehicle and inertia, l is the distance from the center of mass to the rotor shaft, κ_b is the thrust factor and ω_j $j = 1, \ldots, 4$ is the motor speed, $g = 9.81m/s^2$ is the acceleration due to gravity.

The control inputs are given as follows:

$$\begin{cases} T = \kappa_b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ u_1 = \kappa_b(\omega_2^2 - \omega_4^2) \\ u_2 = \kappa_b(\omega_3^2 - \omega_1^2) \\ u_3 = \kappa_\tau(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases}$$

Remark 1. The Euler angles roll and pitch are assumed to be limited to $-\pi/2 < \phi < \pi/2$, $-\pi/2 < \theta < \pi/2$. This assumption is common in practice since the quadrotor vehicle does not perform aggressive maneuvers over free flight.

2.2 Actuator dynamic

A quadrotor UAV actuator system is typically consist of a LiPo battery, a brushless direct current (BLDC) motor and an control stage to control the angular velocity (RPM) of the motor. Electrical DC motors are well modeled by a circuit containing a resistor, inductor, and voltage generator in series [11].

$$v(t) = Ri(t) + L\frac{\partial i(t)}{\partial t} + \frac{\omega(t)}{k_v} \tag{2}$$

where R is the motor internal resistance, L is the inductance, $\omega(t)$ is the rotational rat of the motor, and k_v is the voltage constant of the motor, expressed in $rad/s/volt$. Also, the motor torque τ can be modeled as being proportional to the current $i(t)$ through the torque constant, k_t, expressed in Nm/A.

$$\tau(t) = k_t i(t) \tag{3}$$

The motor dynamics are modeled as a simple first order differential equation (4) where $\dot{\omega}$ is driven by the motor torque and the load friction torque $Q_f(\omega(t))$. The inertia, J includes the motor and the propeller, the motor torque comes from the voltage generator, and the load friction torque results from the propeller drag $Q_f(\omega(t)) = \kappa_\tau\omega^2(t)$, κ_τ is the drag coefficient.

$$J\frac{\partial\omega(t)}{\partial t} = \tau(t) - Q_f(\omega(t)) - D_v\omega(t) \tag{4}$$

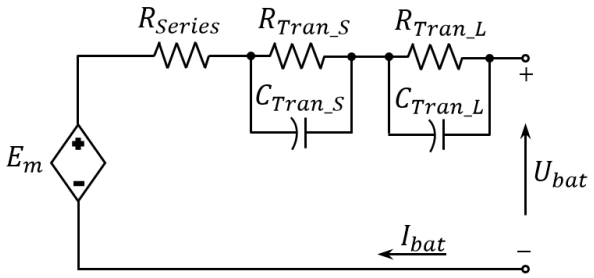


Figure 2: Battery scheme

where D_v is the viscous damping coefficient of the motor Nms/rad. Typically, the inductance of small, DC motors is neglected compared to the physical response of the system and so can be ignored. Under steady-state conditions, the current $i(t)$ is constant, and equation (2) reduces to :

$$v(t) = Ri(t) + \frac{\omega(t)}{k_v} \tag{5}$$

where the term $\frac{1}{k_v}\omega(t)$ represent the electromotive force of the motor. Table 1 shows the motor coefficients for the BLDC motor used in our study.

Parameter	Value
$J\ (kg.m^2)$	$4.1904e^{-5}$
$k_t\ (N.m/A)$	$0.0104e^{-3}$
$k_v\ (rad/s/volt)$	96.342
$D_v(Nms/rad)$	$0.2e^{-3}$
$R\ (Ohm)$	0.2

Table 1: Motor Coefficients

2.3 Battery dynamic

Li-ion battery possesses the greatest potential for future development and optimization. In addition to small size and low weight the Li-ion batteries offer the highest energy density and storage efficiency close to 100%, which makes them ideally suited for portable devices. But the major drawbacks of this type are its high cost [12] [13].

A physical Li-ion battery model was presented. The battery model was designed to accept inputs for current. The outputs were voltage and state of charge (SOC). This model does not take into account the influence of temperature and the phenomenon of self-discharge [12]. However, it gives results close to reality. The model is based on the two equations, the state of charge (SOC) and The voltage across the cell.

$$SOC = 100\left[1 - \frac{\int I}{Q}\right] \tag{6}$$

$$U_{bat} = E_m - R_{int}I \tag{7}$$

E_m is the open circuit voltage (VOC). Its expression is as follows:

$$E_m = E_0 - K\left[\frac{Q}{Q - \int I}\right] + Ae^{-B\int I} \tag{8}$$

E_0 is the open circuit voltage at full load, this is different from the nominal voltage given by the manufacturers. Q is the cell capacity in Ah, The parameters K bias voltage, A exponential voltage and B exponential capacity are experimental parameters determined from discharge curve.

Parameter	Value
$Q\ (Ah)$	1.55
$R_{int}\ (Ohm)$	0.02
$E_0\ (volt)$	1.24
$K\ (volt)$	$2.92e^{-3}$
A	0.156
B	2.35

Table 2: Battery parameters

The model is nonlinear, it was necessary to adapt the input current to that seen by an elementary cell by dividing to parallel branch number $M_{bat} = 1$. For output voltage of our battery, we simply multiplied the output voltage of the cell by the series branch number $N_{bat} = 3$. This methodology requires hypothesize that the cells have the same behavior.

2.4 Energy and motor efficiency

Firsts let define the energy consumed by the vehicle during the mission.

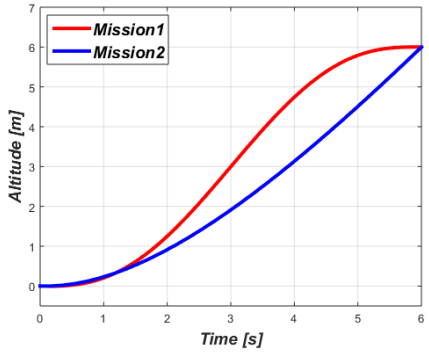
$$E_c = \int_{t_0}^{t_f} \sum_{j=1}^{4} \tau_j(t)\omega_j(t)dt \tag{9}$$

with $\tau_j(t)$ is the torque generated by motor j and $\omega_j(t)$ is rotor speed at time t. By using equation (4) for the four motors, equation (9) can be rewrite as follow:

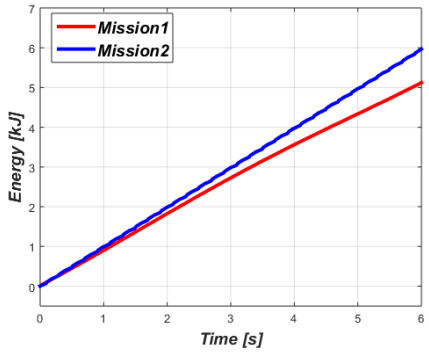
$$E_c = \int_{t_0}^{t_f} \sum_{j=1}^{4} \Big(J\dot{\omega}_j(t) + \kappa_\tau\omega_j^2(t) + D_v\omega_j(t)\Big)\omega_j(t)dt \tag{10}$$

In order to make our energetic model more realistic, an efficiency function is identified and added to energy function (9). The efficiency of the brushless dc motor used for actuate quadrotor helicopter is function of motor torque and rotor speed $f_r(\tau(t), \omega(t))$. We have used polynomial interpolation for efficiency function identification, thus $f_r(\tau(t), \omega(t))$ can be formulated as follow:

$$\begin{aligned} f_r\big(\tau(t), \omega(t)\big) = a\big(\omega(t)\big)\tau^3(t) + b\big(\omega(t)\big)\tau^2(t) \\ + c\big(\omega(t)\big)\tau(t) + d\big(\omega(t)\big) \end{aligned} \tag{11}$$

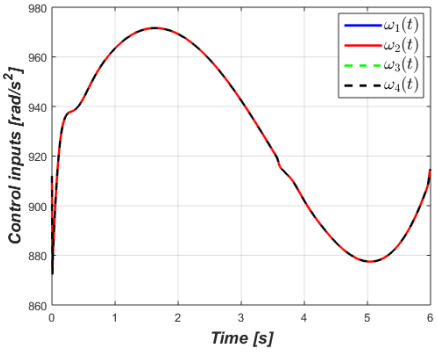


(a)

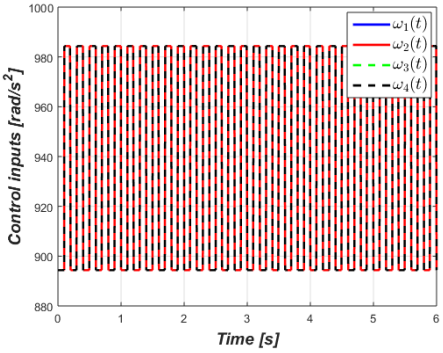


(b)

Figure 3: Quadrotor altitude (a) and consumed energy (b)



(a)



(b)

Figure 4: Control inputs, mission1 (a) mission2 (b)

and

$$a(\omega(t)) = a_1\omega^2(t) + b_1\omega(t) + c_1$$
$$b(\omega(t)) = a_2\omega^2(t) + b_2\omega(t) + c_2$$
$$c(\omega(t)) = a_3\omega^2(t) + b_3\omega(t) + c_3$$
$$d(\omega(t)) = a_4\omega^2(t) + b_4\omega(t) + c_4$$

The parameters of the polynomial are calculated using Matlab for the four motors (we assume that the motors are identical).

$$a_1 = -1.72\ 10^{-5} \qquad b_1 = 0.014 \qquad c_1 = -0.8796$$
$$a_2 = 1.95\ 10^{-5} \qquad b_2 = -0.0157 \qquad c_2 = 0.3385$$
$$a_3 = -6.98\ 10^{-6} \qquad b_3 = 5.656\ 10^{-3} \qquad c_3 = 0.2890$$
$$a_4 = 4.09\ 10^{-7} \qquad b_4 = -3.908\ 10^{-4} \qquad c_4 = 0.1626$$

Then the consumed energy (10) can be rewrite as

$$E_c = \int_{t_0}^{t_f} \sum_{j=1}^{4} \frac{\left(J\dot{\omega}_j(t) + \kappa_\tau\omega_j^2(t) + D_v\omega_j(t)\right)}{f_{r,j}\left(\tau_j(t), \omega_j(t)\right)}\omega_j(t)dt \tag{12}$$

2.5 *Effect of control inputs on energy consumption*

In order to validate the proposed energetic model, we have simulated tow vertical take-off mission with two different altitude trajectory as depicted in Fig. 3-a, and different control inputs as shown in Fig. 4. The missions have the same duration $t_f = 6s$ and the same initial and final configuration.

We have calculate the consumed energy for two mission using equation (12). For the first mission (red line) the consumed energy is $E_1 = 5.12kJ$, where in second mission (blue line) the consumed energy is $E_2 = 5.97kJ$; thus the gain between two mission with the same initial and final configuration is 17%. For a specific mission, energy optimization ca be achieved; what we need is to find control inputs and trajectory that give optimal consumption of energy.

3 Energy Optimization

The energy optimization problem seeks to find control inputs and trajectory for quadrotor helicopter that minimize the consumed energy while satisfying a set of constraints on states and control inputs. In view of optimal control, we try to compute an open-loop solution to an optimal control problem. We looking for control inputs of system (1) that minimize the

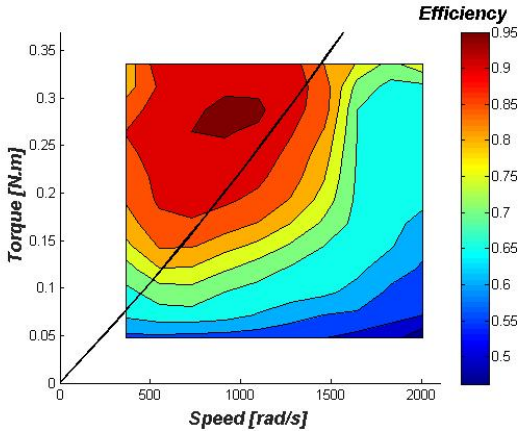


Figure 5: Brushless motor efficiency

consumed energy during a mission between two specific initial and final configurations.

3.1 Problem statement

Now the problem of optimal energy trajectory planning can be formulated as a minimization problem, by which the final consumed energy $E_c(tf)$ is used as the cost function. In addition the state variables in $[x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T$ and control variables in $[\omega_1, \omega_2, \omega_3, \omega_4]^T$ are constrained to satisfy the vehicle dynamics (1) and boundary conditions. The mission is to fly between specified initial and final positions during a time interval $[t_0, t_f]$ where t_0 and t_f are given.

Based on the above description, the optimal control problem can be formulated as:

$$\min_{(\omega_j, \tau_j)} E_c(t_f) \tag{13}$$

subject to

$$\begin{aligned} m\ddot{x} &= (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)T \\ m\ddot{y} &= (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)T \\ m\ddot{z} &= (\cos\phi \cos\theta)T - mg \\ I_x\ddot{\phi} &= (I_y - I_z)\dot{\theta}\dot{\psi} + J\dot{\theta}\varpi + lu_1 \\ I_y\ddot{\theta} &= (I_z - I_x)\dot{\phi}\dot{\psi} - J\dot{\phi}\varpi + lu_2 \\ I_z\ddot{\psi} &= (I_x - I_y)\dot{\phi}\dot{\theta} + u_3 \end{aligned} \tag{14}$$

and

$$\begin{aligned} |\phi| \leq \frac{\pi}{2},\ |\theta| \leq \frac{\pi}{2},\ |\dot{\psi}| \leq \dot{\Psi} \\ \omega_{min} \leq \omega_j \leq \omega_{max} \\ 0 \leq T \leq T_{max},\ |u_k| \leq u_{max},\ k = 1, 2, 3 \end{aligned} \tag{15}$$

with boundary conditions:

$$\begin{aligned} [x(t_0), y(t_0), z(t_0), \phi(t_0), \theta(t_0), \psi(t_0)]^T = \\ [x_0, y_0, z_0, \phi_0, \theta_0, \psi_0]^T \\ [x(t_f), y(t_f), z(t_f)]^T = [x_f, y_f, z_f]^T \end{aligned} \tag{16}$$

The additional constraints in (16) are associated with vehicle dynamics where ω_{min} and ω_{max} are the minimum and maximum feasible velocity of the aircraft rotors, respectively. The roll and pitch angles, $\phi,\ \theta$, have to satisfy $|\phi| \leq \frac{\pi}{2},\ |\theta| \leq \frac{\pi}{2}$ based on their physical definition, and $|\dot{\psi}| \leq \dot{\Psi}$ is required to generate a smooth trajectory where $\dot{\Psi}$ is the maximum changing rate of the heading angle.

4 Nonlinear programming and simulation results

4.1 Nonlinear programming method

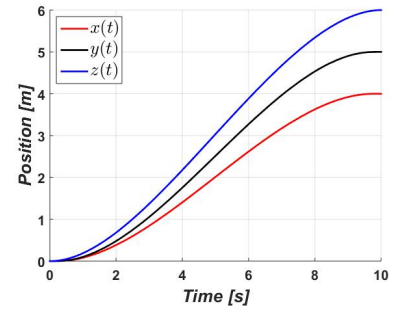
The optimal control problem presented in the previous sections (13)-(17) is a complex nonlinear optimization problem. The general approach to solve this problem is the direct collocation method. The basic idea of direct collocation is to discretize a continuous solution to a problem represented by state and control variables by using linear interpolation to satisfy the differential equations. In this way an optimal control problem is transformed into a nonlinear programming problem (NLPP).

In our study, the proposed optimal control problem have been numerically solved using a Matlab software called GPOPS-II [14]. The software employs a Legendre-Gauss-Radau (LGR) [15][16] quadrature orthogonal collocation method where the continuous-time optimal control problem is transcribed to a large sparse nonlinear programming problem (NLP). it used an adaptive mesh refinement method that determines the number of mesh intervals and the degree of the approximating polynomial within each mesh interval to achieve a specified accuracy. The software allows the use of two nonlinear programming (NLP) solver used to solve the NLPP. The first is the open-source NLP solver IPOPT (Interior Point OPTimizer) [17], where the second is the NLP solver SNOPT (Sparse Nonlinear OPTimizer) [18].

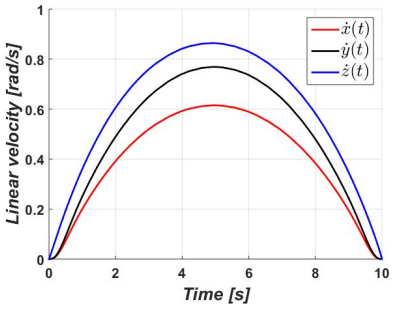
4.2 Simulation Results

Problem (13)-(17) was solved using the open-source NLP solver IPOPT in second derivative (full Newton) mode with the publicly available multi-frontal massively parallel sparse direct linear solver MUMPS [19]. All results were obtained using the implicit integration form of the Radau collocation method and various forms of the aforementioned *ph* mesh refinement method using default NLP solver settings and the automatic scaling routine in GPOPS-II.

In our tests we considered the DJI Phantom 2 quadrotor [20] with multi-rotor propulsion system (2212/920KV motors). The physical parameters of the Phantom 2 used in the simulation experiment, are reported in Table 3.

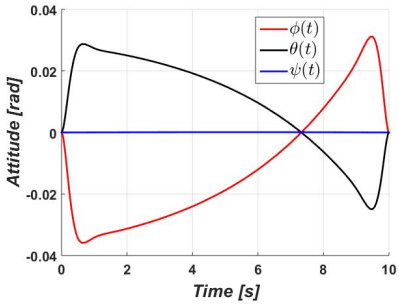


(a)

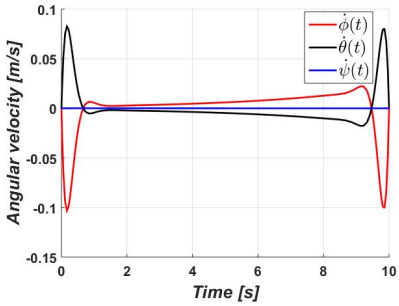


(b)

Figure 6: Quadrotor position (a) and linear velocity during the trajectory (b)



(a)



(b)

Figure 7: Quadrotor attitude (a) and angular velocity (b)

The problem (13)-(17) was numerically solved to find the minimum energy control inputs $\omega_j(t)$ that allows quadrotor to fly from the initial position $[x, y, z]^T = [0, 0, 0]^T$ at time $t_0 = 0$ to the final one $[x, y, z]^T = [4, 5, 6]^T$ at time $t_f = 10s$, with initial condition $[x_0, y_0, z_0, \phi_0, \theta_0, \psi_0]^T = [0_{1\times6}]^T$ and final condition $[x_f, y_f, z_f, \phi_f, \theta_f, \psi_f]^T = [4, 5, 6, 0, 0, 0]^T$. Null initial angular and linear velocities were considered $[\dot{x}_0, \dot{y}_0, \dot{z}_0, \dot{\phi}_0, \dot{\theta}_0, \dot{\psi}_0]^T = [0_{1\times6}]^T$ and the same for final angular and linear velocities . With respect to constraints in (12) the initial guess for inputs control is given by $w_s = 912,32\ rad/s$ which means that the total thrust is $T = 12,75\ N$ which corresponds the thrust necessary to counterbalance the gravity acceleration. Fig. 6 shows the time evolutions of the vehicle position and linear velocity, were Fig. 7 shows the time evolutions of the vehicle attitude and angular velocity. Fig. 8 reports The optimal trajectory in $(x - y - z)$ space and the control inputs $\omega_j(t)$. The energy consumed by the quadrotor to ravel this trajectory is $E_c(t_f) = 10.38\ kJ$ Fig. 9.

4.3 Comparative study

In order to evaluate the energy consumed by the vehicle and have an idea about the saving energy with the proposed approach, we have compared the energy consumed by the quadrotor vehicle under our optimal control approach and the energy consumed under a classical control approach.

The classical control approach consist of two controller, a low-level controller and the high-level control algorithm for quadrotor vehicle.

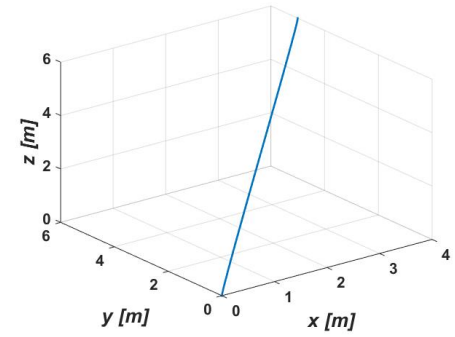
The low-level control module consist of an adaptive fuzzy backstepping controller [10], composed of three terms , the fuzzy adaptive control term which is designed to approximate a model-based backstepping control law $u_{f,j} = \Theta_j^T \varphi_j(x_s)$, where Θ_j^T is the vector of fuzzy basis functions, $\varphi_j(x_s)$ is the vector of adjustable parameters of the fuzzy logic system and x_s is the state vector. The bounded robust control term $u_{r,j} = \hat{\delta}_j \tanh(\frac{e_{2j}}{\epsilon_j})$ employed to compensate the fuzzy approximation error. Finally, $u_{p,j} = k_{2j} e_{2j}$ the proportional derivative term.

The tracking errors is defined as:

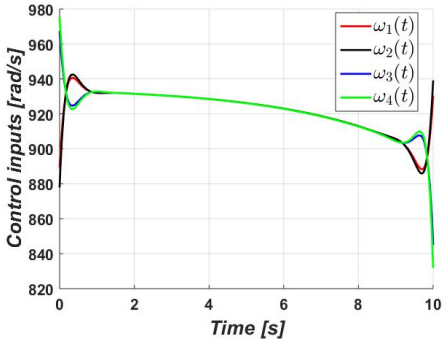
$$e_{1j} = x_{1j,d} - x_{1j}\ ,\ j = 1, \ldots, 6 \tag{17}$$

where $x_{1j,d}$ is the position and attitude desired signals. Using position controller u_5 and u_6 (18), the desired roll and pitch signals can be calculated as $\theta_d = atan(\frac{u_5 \cos\psi + u_6 \sin\psi}{g})$, $\phi_d = atan(\frac{u_5 \sin\psi - u_6 \cos\psi}{g} \cos\theta_d)$. The backstepping second tracking errors signals is defined as

$$e_{2j} = \upsilon_j - x_{2j} \tag{18}$$



(a)



(b)

Figure 8: The optimal trajectory in $(x-y-z)$ space (a), and the control inputs (b)

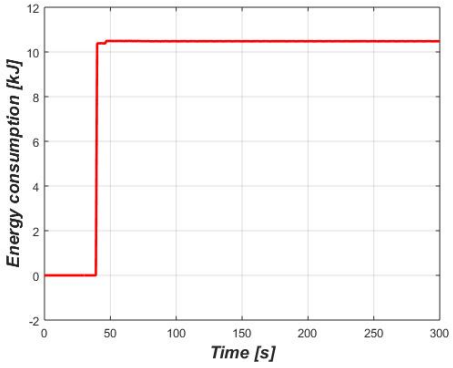


Figure 9: Cost function $E_c(t)$

with the virtual control law $\upsilon_i = \dot{x}_{1j,d} + k_{1j}e_{1j}$ and $k_{1j} > 0$. Then, we introduce the following tracking control algorithm.

$$u_j(t) = \Theta_j^T \varphi_j(x_s) + \hat{\delta}_j \tanh\left(\frac{e_{2j}}{\epsilon_j}\right) + k_{2j}e_{2j} \qquad (19)$$

$$\dot{\Theta}_j = \gamma_j e_{2i} \varphi_j(x_s) \qquad (20)$$

$$\dot{\hat{\delta}}_j = \eta_j e_{2j} \tanh\left(\frac{e_{2j}}{\epsilon_j}\right) \qquad (21)$$

where $k_{2j} > 0, \epsilon_j > 0, \gamma_j > 0, \eta_j > 0$ are design parameters.

The high-level control algorithm consist of a third degree polynomial, $q(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3$, with initial and final conditions on position and velocity $q(t_0) = q_0,\ q(t_f) = q_f,\ \dot{q}(t_0) = \dot{q}_0,\ \dot{q}(t_f) = \dot{q}_f$ we can calculate parameters of the polynomial.

4.4 Comparison

The consumed energy obtained with the classical approach is $E_c(t_f) = 10.49\ kJ$, compared to the energy consumed by the proposed approach, its increase with 1% from the total energy, which is not a good saving quantity of energy. To improve the saving quantity of energy we must make he model more energetic, which will be the objective of future work.

We have made a comparative study between our approach and the approach proposed by [9] for a final time $t_f = 20s$. The consumed energy obtained with the approach proposed in [9] is $E_{c1}(t_f) = 26.23\ kJ$, where the energy consumed by our approach for the same boundary conditions is $E_{c2}(t_f) = 20.72\ kJ$. The energy saved by the proposed approach compared to the one proposed in [9] is 26.59% from the total energy.

Parameter	Value
$l\ (m)$	0.175
$m\ (kg)$	1.3
$I_x\ (kgm^2)$	0.081
$I_y\ (kgm^2)$	0.081
$I_z\ (kgm^2)$	0.142
$\kappa_b\ (N/rad/s)$	$3.8305e^{-6}$
$\kappa_\tau\ (Nm/rad/s)$	$2.2518e^{-8}$

Table 3: Quadrotor parameters

5 Conclusion and Future Work

In this paper we have introduced an energetic model for quadrotor vehicle. The energetic model contains vehicle movement dynamic, actuators dynamic, battery dynamic and an efficiency function for energy computing. Then, we have validated the energetic model through two different mission with the same initial and final configurations. For energy optimization purpose, an optimal control problem have been introduced and solved using an optimal control software GPOPS-II. In the optimal control problem we seeks to find the vehicle control inputs and trajectory that minimize the consumed energy during a specific mission. The numerical experiments illustrated the solutions of the proposed optimal control problem, and the comparative study provide quantization of energy that can be saved in a simple mission.

In future work, we will incorporate propeller aerodynamic to improve the energetic model, and we will introduce an energy optimization problem with respect to battery life. We are also going to advise an experimental procedure to validate the proposed approach.

REFERENCES

[1] Kurt A Swieringa, Clarence B Hanson, Johnhenri R Richardson, Jonathan D White, Zahid Hasan, Elizabeth Qian, and Anouck Girard. Autonomous battery swapping system for small-scale helicopters. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 3335–3340, 2010.

[2] Jeremie Leonard, Al Savvaris, and Antonios Tsourdos. Energy management in swarm of unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, 74(1-2):233–250, 2014.

[3] N Kemal Ure, Girish Chowdhary, Tuna Toksoz, Jonathan P How, Matthew A Vavrina, and John Vian. An automated battery management system to enable persistent missions with multiple aerial vehicles. *Mechatronics, IEEE/ASME Transactions on*, 20(1):275–286, 2015.

[4] James F Roberts, Jean-Christophe Zufferey, and Dario Floreano. Energy management for indoor hovering robots. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 1242–1247, 2008.

[5] Analiza Abdilla, Arthur Richards, and Stephen Burrow. Power and endurance modelling of battery-powered rotorcraft. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 675–680, 2015.

[6] Michal Podhradskỳ, Calvin Coopmans, and Austin Jensen. Battery state-of-charge based altitude controller for small, low cost multirotor unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, 74(1-2):193–207, 2014.

[7] D. C. Gandolfo, L. R. Salinas, A. Brando, and J. M. Toibero. Stable path-following control for a quadrotor helicopter considering energy consumption. *IEEE Transactions on Control Systems Technology*, PP(99):1–8, 2016.

[8] Kevin Vicencio, Tristan Korras, Kenneth A Bordignon, and Iacopo Gentilini. Energy-optimal path planning for six-rotors on multi-target missions. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 2481–2487, 2015.

[9] Fabio Morbidi, Roel Cano, and David Lara. Minimum-energy path generation for a quadrotor uav. In *Robotics and Automation (ICRA), IEEE International Conference on*, 2016.

[10] Fouad Yacef, Omar Bouhali, Mustapha Hamerlain, and Nassim Rizoug. Observer-based adaptive fuzzy backstepping tracking control of quadrotor unmanned aerial vehicle powered by li-ion battery. *Journal of Intelligent & Robotic Systems*, 84(1):179–197, 2016.

[11] Mark Cutler, N Kemal Ure, Bernard Michini, and Jonathan P How. Comparison of fixed and variable pitch actuators for agile quadrotors. In *AIAA Conference on Guidance, Navigation and Control*, 2011.

[12] Tedjani Mesbahi, Nassim Rizoug, Patrick Bartholomeus, and Philippe Le Moigne. Li-ion battery emulator for electric vehicle applications. In *2013 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1–8. IEEE, 2013.

[13] KC Divya and Jacob Østergaard. Battery energy storage technology for power systemsan overview. *Electric Power Systems Research*, 79(4):511–520, 2009.

[14] Michael A. Patterson and Anil V. Rao. Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Trans. Math. Softw.*, 41(1):1:1–1:37, October 2014.

[15] Divya Garg, William W. Hager, and Anil V. Rao. Pseudospectral methods for solving infinite-horizon optimal control problems. *Automatica*, 47(4):829 – 837, 2011.

[16] Michael A Patterson and Anil Rao. Exploiting sparsity in direct collocation pseudospectral methods for solving optimal control problems. *Journal of Spacecraft and Rockets*, 49(2):354–377, 2012.

[17] Lorenz T Biegler, Omar Ghattas, Matthias Heinkenschloss, and Bart van Bloemen Waanders. Large-scale pde-constrained optimization: an introduction. In *Large-Scale PDE-Constrained Optimization*, pages 3–13. Springer, 2003.

[18] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

[19] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[20] DJI Phantom 2. http://www.dji.com/fr/phantom-2. Accessed: 2016-02-25.

Development and Design Methodology of an Anti-Vibration System on Micro-UAVs

Zhenming Li[1], Mingjie Lao[2], Swee King Phang[2]*, Mohamed Redhwan Abdul Hamid[2], Kok Zuea Tang[3], and Feng Lin[2]
[1]Mechanical Engineering, National University of Singapore, Singapore
[2]Temasek Laboratories, National University of Singapore, Singapore
[3]Electrical and Computer Engineering, National University of Singapore, Singapore

ABSTRACT

As the potential applications of unmanned aerial vehicles (UAVs) are growing, more sensors are installed on-board. Mechanical vibration of the UAV, which greatly hinders the accuracy of its on-board sensors, becomes an increasingly important issue. In this manuscript, an anti-vibration framework on micro-UAVs is proposed. The vibration sources of the UAV will be investigated and identified. Then, several selections of hardware dampers are tested along with a digital low-pass filter on actual UAV. With the results from different case studies, a criteria of damper selection for micro-UAV is built to serve as a guideline for further practice.

1 INTRODUCTION

An unmanned aerial vehicle (UAV), commonly known as a drone, may be controlled by a remote-pilot or operated autonomously without human intervention. Due to its small size and remoteness during operation, modern miniature-UAVs are now widely used in both commercial and military aspects to fulfill the needs in different scenarios [1]. One of the popular research areas is on scouting of remote areas with limited GPS reception (such as indoor or foliage environment) through on-board cameras for localization mean [2].

To realized such applications, useful data are often collected from the sensors such as inertial measurement unit (IMU) and camera images at low frequencies (1-20 Hz approximately) to realize navigation or localization algorithms by estimating the UAV position and bearing [3, 4, 5]. The accuracy of these data is important for the flight control system to ensure good performance during flights. One major factor hindering the accuracy of these data is the mechanical vibration of the UAV during flights. Among the multiple causes to the vibration problem, rotating rotors and structure vibration on its natural frequencies are of most concern [6, 7].

Fortunately, these vibrations are usually recorded as higher frequency signals by IMU and they tend to pollute the useful signals at low frequency through the phenomenon of aliasing. Under such effect, higher frequency signals are observed as lower frequency signals causing inaccurate data. To avoid aliasing effect, one common method is to sample the data with the rate of at least 2 times of the maximum frequency of the signal. However, there is a limit to how fast the sampling frequency is and the relative high frequency of vibration generated signals will continue to impair the accuracy of the useful signals. Insufficient damping to such vibrations will also likely result in sensor drifting such as coning and sculling motion which is the condition where the output received by gyroscope or accelerometer becomes more inaccurate over time.



Figure 1: T-Lion developed at TL@NUS

Acknowledging the importance of implementing anti-vibration measures on-board, different dampers and isolators have been considered and recommended for vibration minimization. For instants, Kyosho Zeal sheet and its performances have been briefly studied in [8]; Wire-rope isolators and rubber dampers for minimum vibration effects towards data logging during flight has been discussed in [9]. In this manuscript, theoretical research will be done on a selections of dampers and isolators for the best anti-high-frequency vibrations to be used in UAV. The results will be further verified by state-of-the-art bench tests and UAV flight trials.

The UAV to be used to verify the proposed anti-vibration system is an in-house UAV developed by the Temasek Laboratories at the National University of Singapore (TL@NUS), codenamed T-Lion (see Fig. 1). It weighs 3 kg and is capable to carry additional 2 kg payload. On-board system to be isolated from the UAV structural vibration consists of a full IMU sensor suite weighs 120 g.

The manuscript is divided as follows: Section 1 will be

*Email address: skphang@nus.edu.sg

given as introduction to this work; Section 2 lists the relevant theoretical frameworks involved in the study; Section 3 discusses the sources of vibrations in UAV; case studies of selected dampers are examined in Section 4; results verification via flight trials will be shown in Section 5 while concluding remarks will be made in the last Section.

2 Theoretical Frameworks

2.1 Vibration Damping and Vibration Isolation

Isolators are used to lower the natural frequency of the system to below the excitation (or disturbing) frequency (which in this case, the vibration frequency created by motors). Ultimately, the aim is to keep these two frequencies out-of-sync by 180° so that to avoid effects such as resonance. On the other hand, dampers are used to remove mechanical energy from disturbing vibration out of the system by absorbing the energy and converting to other forms of energy such as heat [10]. Normally, anti-vibration devices available in the market are both isolators and dampers. For simplicity, these are referred as dampers in this paper.

2.2 Transmissibility Curve

For effective vibration isolation from undesired high frequency, damper would need to have a natural frequency at least less than 50% of the lowest disturbing frequency, and optimally, less than 71%. In other words, the ratio of disturbing frequency over natural frequency is more than $\sqrt{2}$. This can be explained by transmissibility which is defined as the ratio of force transmitted through the suspension apparatus to force applied by vibration and it can be calculated through

$$T = \left|\frac{A_o}{A_i}\right| \equiv \left|\frac{a_o}{a_i}\right| \equiv \left|\frac{F_o}{F_i}\right|, \quad (1)$$

where A_o and A_i are the amplitudes for output and input respectively, a_o and a_i are the accelerations for output and input respectively, F_o and F_i are the forces for output and input (applied and transmitted) respectively [11].

Vibrations can never be completely removed from the system, i.e., $T \neq 0$ and hence minimizing this ratio for a specific frequency range is the goal of vibration isolation. Transmissibility curve can be divided into three regions: no effect region ($T = 1$), amplification region ($T > 1$), and isolation region ($T < 1$). In general, it is highly desirable to have the natural frequency of the damper as far apart as possible from the disturbing frequency to achieve effective vibration isolation. In general, isolation efficiency increases as the transmissibility decreases along increasing system frequency.

2.3 Static Deflection

Static deflection is how much the damper deflects when it is subjected to the static weight of the equipment it carries. In general, the larger the static deflection that the damper has before damping process, the better isolation effect can be achieved. In a single degree-of-freedom (DoF) system, natural frequency and static deflection are estimated to have an inverse linear relationship; larger static deflection will have a lower system natural frequency, and vice versa. As mentioned earlier, lower natural frequency leads to a larger separation from the disturbing frequency, thus better isolation efficiency. Therefore, it is important to use soft, flexible dampers for light damping mass to ensure a fair amount of static deflection is present.

2.4 Damper Selection Criteria

Judging from the above-mentioned characteristics, a few essential criteria for selecting dampers to be used in our studies are

1. Electrical insulator to avoid short-circuit;
2. Soft and flexible;
3. Natural frequency outside UAV structural resonance zone;
4. Low compression set and low creep;
5. Good resistance to outdoor conditions; and
6. Easy installation and adjustment.

3 Source of Vibration

Two main sources of high amplitude vibrations come from the rotating rotors, which has the same frequency to the rotating rate, and structural natural frequency vibrations, which the frequencies depend on the UAV structure. For T-Lion UAV, the rotors rotates at approximate 50 revolutions per second, which translates to 50 Hz vibration on the UAV. This section will be divided into two parts, where the structural natural frequencies will be studied in a simulation and the overall vibrations over a large frequency band will be obtained through experimental data with actual flight.

3.1 Structural Vibration Analysis

The T-Lion UAV is modeled in SolidWorks simulation, with the actual material properties assigned to each part of the simulated model. Frequency analysis on the model was carried out and vibration frequencies with significant impact are recorded as follows:

1. There are obvious high magnitude vibrations on x- and y-directions at 39.90 Hz (see Fig. 2 for mode shape visualization), and on z-direction at 80.48 Hz. From the simulation result, they are mainly caused by the hanging payload of T-Lion UAV below the UAV central region. It is also observed that similar behaviour also exists at 160.17 Hz and 321.82 Hz, which strongly suggested that they are the 3rd and 4th mode natural frequencies of the payload; and
2. Several small amplitude vibrations are observed between 100 to 200 Hz with lesser than 1 mm amplitude

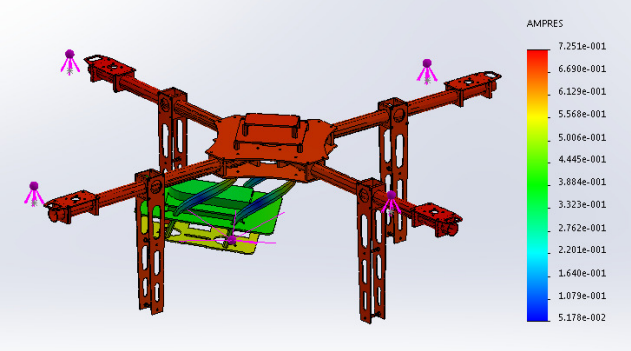


Figure 2: Mode shape of T-Lion at 39.9 Hz

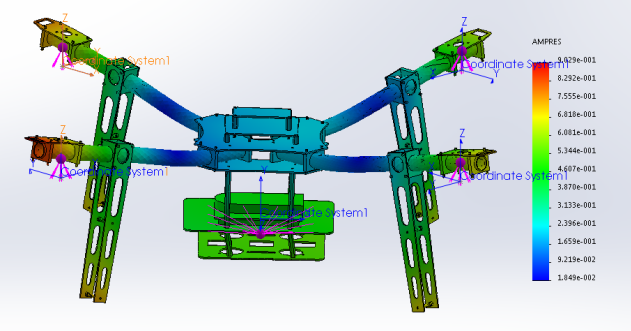


Figure 3: Mode shape of T-Lion at 109.88 Hz

(see Fig. 3 for mode shape visualization). A strong vibration of 3.15 mm amplitude is also observed at frequency of 273.16 Hz. They are most likely the structural natural frequencies of the UAV due to extending arms and components.

3.2 Actual Flight Data

To verify the vibration frequency of the UAV structure, an actual flight trial was carried out where the T-Lion was commanded to hover for several seconds in the air. An additional IMU sensor, the ADIS IMU was installed to the UAV for higher sampling rate of more than 800 Hz. With this sensor, acceleration data can be collected and converted to frequency domain with a frequency range up to 400 Hz. Experiments are designed to identify and examine different vibration signals and their corresponding vibration locations of the UAV.

Result in Fig. 4 shows that there is significant amount of vibration signals at high frequencies between 40 Hz to 400 Hz. Their magnitudes are large and thus cannot be neglected. By comparing the frequency response of the actual system with the simulated results above, vibrations of the UAV can be divided into 3 parts, demarcated clearly in Fig. 4:

1. A: These low frequency vibrations (approximately 40 Hz at x- and y-directions, approximately 80 Hz at z-direction) correspond to the vibrations at the payload of the UAV where the battery and certain weight source were located at the bottom of the UAV;
2. B: Vibration signals at B is originated from the rotating rotors at approximate 50 Hz; and
3. C: Vibration of the structure of UAV, mainly on the platform and the extension arms.

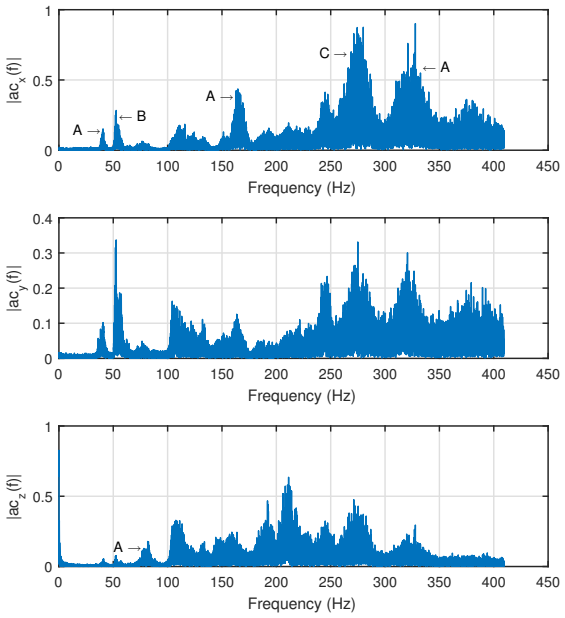


Figure 4: Vibration of T-Lion UAV measured by ADIS IMU

These vibration frequency peaks are rather consistent to our simulated results. It is concluded that structural vibrations are mainly high frequency signals, and some of them are so high that low-cost IMU sensor (with sampling rate below 100 Hz) would not detect them. They are undesired and thus dampers will be designed to filter them.

4 Case Studies

According to the damper selection criterion discussed in Section 2, four dampers have been selected for performance evaluations. The dampers can be visualized in Fig. 5.

4.1 Silicone Ball Damper

Silicone ball damper has similar functionality to the rubber damper, but it is made by better material. In general, silicone damper is softer than rubber damper leading to larger static deflection, which is preferable to reduce vibrations. However, it is difficult to install and to make changes to the damper on UAV due to its complicated installation mechanism. Furthermore, it requires high damping weight ($>$ 200 g) for efficient damping while the damping mass for T-Lion

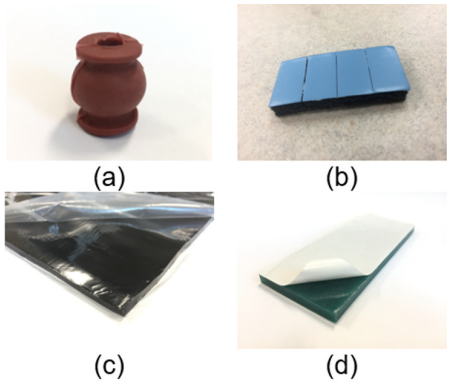


Figure 5: (a) Silicon Ball; (b) Silicon Foam; (c) Sorbothane sheet; and (d) Kyosho Zeal sheet

UAV is too light for this requirement (120 g). This option is thus discarded and will not be further discussed.

4.2 Silicone Foam

Silicone Foam is the softest and the most flexible damper among all. Therefore, it is possible to damp a light weight IMU alone without the need to add extra mass. However, it may have very low natural frequency and eventually cause resonance with the useful signals at low frequency. On top of that, it breaks easily and thus not suitable to reuse.

4.3 Kyosho Zeal and Sorbothane 30 Durometer Sheets

Kyosho Zeal sheet and Sorbothane 30 Durometer sheet are most used tape dampers recognized by many studies and experiments. They are easy to install and relatively soft to accommodate light masses. These can also be installed by clamping the mass in between two pieces of pads and compressing them to about 80% of its original thickness. This installation method enables the UAV to do inverse flying without compromising on damping. Kyosho Zeal sheet is relatively softer than Sorbothane 30 Durometer sheet, but their performance is comparable. However, these sheets have relatively higher compression set.

4.4 Damper Performances

Using vibration table, static vibration tests have been done to verify the dampers performances subjecting to vibrations for a frequency range from 10 to 300 Hz. The actual loading on T-lion UAV is 120 g in total, which translates to 30 g per damper. Here, the stimulated damping mass used is 30 g to be consistent to the actual UAV load mass. Different dimensions are tested for selected dampers. Fig. 6 shows an example of the set-ups of the system on vibration table.

With Equation 1, transmissibility of each damper can be calculated with output and input acceleration data collected. Transmissibility curve for each damper is obtained by plotting the calculated transmissibilities with its corresponding frequencies. Observed from the results shown in Fig. 7, there

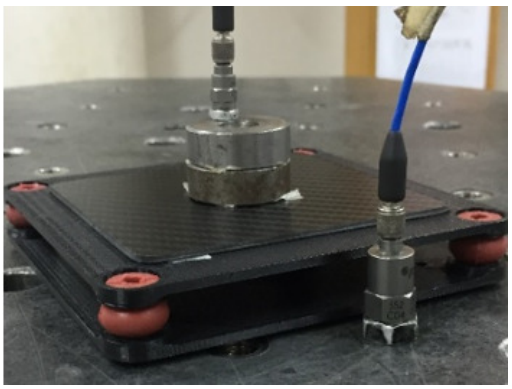


Figure 6: Vibration test setup on vibration table

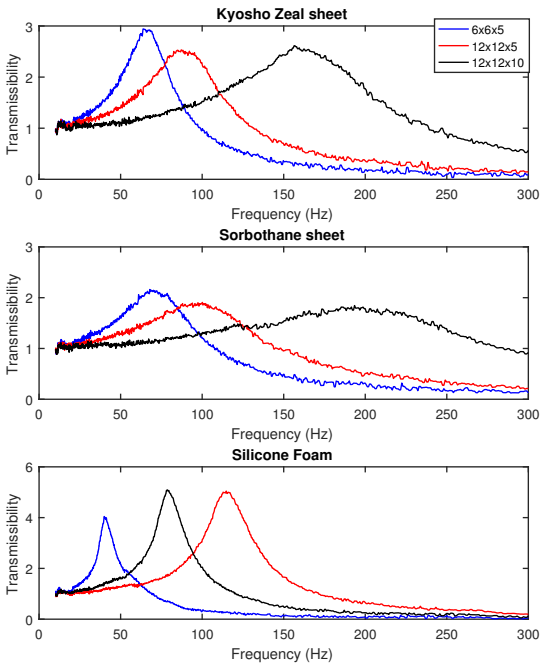


Figure 7: Transmissibility curve with different dimensions for Kyosho Zeal sheet, Sorbothane sheet, and Silicone foam

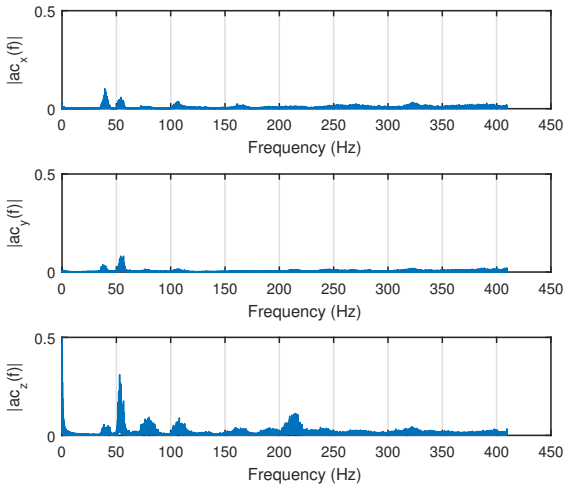


Figure 8: Vibration generated in actual UAV flight recorded by ADIS IMU sensor

is a clear shifts of the transmissibility curve towards the left when the smaller dampers are used, and this results to lower natural frequency of the damper. Ideally, the lower the natural frequency of the damped system, the better the isolation damper provides for system with high disturbing frequency.

Despite Silicone Foam has the lowest natural frequency, its high flexibility makes it sensitive to disturbances which worsens its performance. Overall, Kyosho Zeal Sheet has the best performance among the selections.

Its effect can be further proven by ADIS IMU testing as discussed earlier. Fig. 8 shows that almost all magnitudes of vibrations are reduced after a Kyosho Zeal sheet is installed to damp the sensor. Noted that for the vibrations at higher frequencies, the reduction is up to 96.8%. Also, as expected, motor noise at around 50 Hz is magnified as suggested in the transmissibility curve shown previously. However, this does not affect its effectiveness in minimizing vibrations at higher frequencies.

5 Flight Experiments

As Kyosho Zeal sheet was found to be best suited to damp the on-board sensors of weight 30 g each (120 g in total) from the previous section, the on-board system which includes IMU sensors were mounted on Kyosho Zeal sheet for actual UAV implementations on our in-house quad-copter codenamed T-Lion. In this section, more dimensions and different installation methods of Kyosho Zeal sheet were used and its effect on vibration reduction is studied.

5.1 Hardware Damper Design

Flight tests are conducted with the different configurations in Table 1 aiming to verify the effects of each parameter

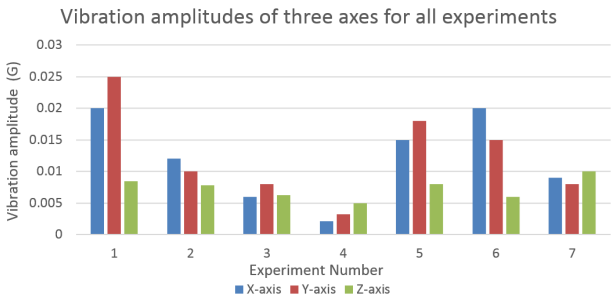


Figure 9: Kyosho Zeal sheet test results on three axes of UAV

Experiment	Dimension (mm)	Installation	Contact
1	$12 \times 12 \times 5$	Single	Less
2	$12 \times 12 \times 5$	Single	More
3	$12 \times 12 \times 10$	Single	More
4	$6 \times 6 \times 5$	Single	More
5	$6 \times 6 \times 5$	Clamped	More
6	$12 \times 12 \times 5$	Clamped	More
7	$12 \times 12 \times 10$	$45°$	More

Table 1: Kyosho Zeal sheet installation parameters

with regarding to Kyosho Zeal sheet performances.

Fig. 9 shows that the variations of vibration amplitudes of x-, y- and z-axis across different configurations of Kyosho Zeal sheets at 50 Hz. According to the results, experiment 4 has the best result in attenuating the vibration signals on T-lion UAV as the vibration amplitude is the lowest at 0.0021 G, 0.0032 G, and 0.005 G, comparing to vibration amplitude without any dampers at 0.08 G, 0.13 G and 0.17 G.

In general, the following guidelines for damper installation provide the best result for UAV vibration reduction:

1. More contact surface area of the damper is preferred;
2. Kyosho Zeal sheet should be installed vertically;
3. Even Length-Width-Height (LWH) ratio ($\approx 1:1:1$) gives the best results; and
4. Damper with smaller dimensions performs better for light mass damping.

5.2 Digital Low-Pass-Filter Design

Actual flight experiment results show significant reductions in the vibration amplitudes (up to 97.1% reduction along the z-axis) with both damper and low-pass filter (cut-off frequency: 10 Hz) implemented. The illustration below (Fig. 10) shows that vibration signals could be amplified with damper alone, but the combined effect of both low-pass filter and damper will shift the system into isolation region, and this justifies the experiment results.

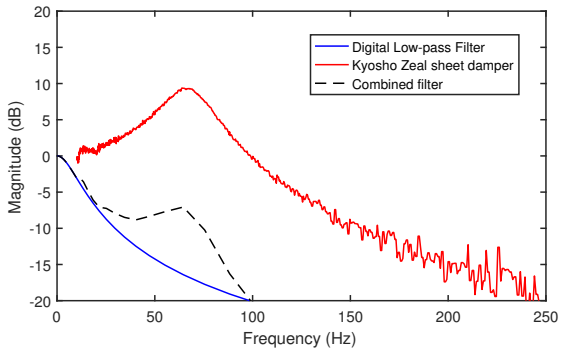


Figure 10: Combined frequency response of hardware damper and digital filter

Despite the high flexibility and softness of the chosen dampers, mechanical dampers show their limitations in damping light masses and at low frequency such as 50 Hz. On another hand, low-pass filter also has its limitation as its functional frequency range is much shorter than mechanical damper. Therefore, the use of low-pass filter is essential to lower amplified signal due to dampers while dampers are used to reject the high frequency vibrations due to the UAV structure.

6 Conclusion

This manuscript elaborates in detail the essential theoretical frameworks when considering anti-vibration measures on UAVs and how these can be used to improve on the current measures. Building on from this, criteria of damper selection for UAVs with light damping mass is established to act as guidelines for future practices.

With close examinations on vibration sources generated on T-lion UAV through both simulation and experimental results, it is shown that structural vibration contributes much higher vibration amplitude at higher frequencies comparing to motor vibration at lower frequency, addressing the importance of considering vibration attenuation for structural vibration when designing anti-vibration measures. Therefore, the relationships between different parameters of dampers and their damping effectiveness are closely evaluated through vibration tests, and the best performer, Kyosho Zeal sheet, further proves its effectiveness in minimizing structural vibrations at higher frequency, despite of creating slight signal amplification for motor vibration. This also indicates the limitation of mechanical damper in damping light mass at low disturbing frequency. Nonetheless, with both digital low-pass filter and Kyosho Zeal sheet, actual flight test shows up to 97.1% improvement on lowering the vibration amplitude. This addresses the significance of implementing both digital filter and mechanical damper for effective damping on UAVs.

Further studies can be done on vibration tests to investigate the pattern of how different parameters change with various masses and dimensions of the dampers in details, so that to predict its general performance and the optimal working range (in terms of damping mass and natural frequency) for each damper.

Lastly, this research work can be extended to other implications such as UAVs with different configurations and helicopter drones by making use of the frameworks and evaluation results provided in this paper.

References

[1] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa. *Autonomous Flying Robots*. Springer, 2010.

[2] F. Wang, S. K. Phang, J. J. Ong, B. M. Chen, and T. H. Lee. Design and construction methodology of an indoor uav system with embedded vision. *Control and Intelligent Systems*, 40(1):22–32, 2012.

[3] S. K. Phang, F. Wang, K. Wang, S. Lai, and B. M. Chen. An effective method for autonomous localization and navigation in unknown indoor environment using mav. In *International Micro Air Vehicles Conference*, 2015.

[4] J. Langelaan and S. Rock. Navigation of small uavs operating in forests. In *AIAA Guidance, Navigation and Control Conference*, 2004.

[5] B. Sinopoli, M. Micheli, G. Donato, and J. T. Koo. Vision based navigation for an unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation*, 2001.

[6] T. H. Lee G. Cai, B. M. Chen. *Unmanned rotorcraft systems*. Springer, London/New York, 2011.

[7] V. Gavrilets, A. Shterenberg, M. A. Dahleh, and E. Feron. Avionics system for a small unmanned helicopter performing aggressive maneuvers. In *Digital Avionics Systems Conference*, 2000.

[8] A. Imam and R. Bicker. Design and construction of a small-scale rotorcraft uav system. *International Journal of Engineering Science and Innovative Technology*, 3(1):96–109, 2014.

[9] G. Cai, F. Lin, B. M. Chen, and T. H. Lee. Systematic design methodology and construction of uav helicopters. *Mechatronics*, 18(10):545–558, 2008.

[10] R. Simmons. Vibration isolation. *ASHRAE Journal*, pages 30–40, 2007.

[11] H. Fonseca. Displacement and force transmissibility in structures and multilayer supports with applications to vibration isolation. Master's thesis, Universidade Tecnica De Lisboa Instituto Superior Tecnico, Lisbon, Portugal, 2011.

Quick Aerodynamic Design of Micro Air Vehicles

V. Vyshinsky, A. Kislovskiy*
Moscow Institute of Physics and Technology, 140180, Gagarin street, 16, Zhukovskiy, Russia
Central Aerohydrodynamic Institute, TsAGI, 140180, Zhukovskiy street, 1, Zhukovskiy, Russia

ABSTRACT

The quick aerodynamic characteristics estimation method for preliminary design phase is presented in the paper. Simplified mathematiclal model of aircraft layout, robust and fast direct CFD code as well as artificial neural network (ANN) technique form the basis of the method. For illustration of its possibilities the proposed method was applied to micro air vehicles (MAV) design. Developed mathematical representation approximates MAV layout with 100-dimension parameter vector. The ranges of mathematical representation parameters (aspect ratio, dihedral angle and area of the wing, airfoil relative thickness, airfoil geometry etc.) were determined during existing MAV market review. The layout generator creates a number of layouts, runs CFD computations on different flight regimes. Then the information about flight regime is appended to input vector of the main algorithm. Calculated aerodynamic characteristics forms the output. Outlying layouts are filter out using geometric and aerodynamic criteria. The resulting set of vectors forms training and test sets for machine learning algorithms. For aerodynamic force and momentum coefficients calculations, the separate ANNs were created.

1 INTRODUCTION

Artificial Neural Nets (ANN) are widely used in aerodynamics and aeronautical engineering nowadays. Recent aerodynamic applications include, for example, flow control, estimation of aerodynamic coefficients, compact functional representations of aerodynamic data for rapid interpolation, grid generation,and aerodynamic design [1]. Some works showed that the preliminary design phase may be significantly simplified and accelerated with new kind of aerodynamic design tools compared with traditional approaches [2, 3, 4].

Quick estimation of MAV aerodynamic characteristics on cruise flight is significant practical problem, especially given their small-batch production, huge variety of models and wide range of their payloads.

The key suggested in the paper are robust CFD code, ANN technique and reasonable choice of layout representation (Figure 1). ANN technology requires huge amount of data for training. Creation of the layout generator is very impotant due to the lack of specific MAV data. The most important feature of the layout generator is creating objects having desired properties [5, 6].

Data generation module was divided into two parts: layout generation and CFD code that calculates aerodynamic cahracteristics (lift, drag and pitching moment).

In the first stage a mathematical representation of MAV layout was developed (which contains 100 parameters). The design process in this case is significantly simplified in comparison with traditional methods.

The flow around the MAV was calculated in the prescribed range of free stream parameters. The results of CFD calculations combined with MAV representation form the dataset.

Machine learning algorithms can be applied to generated data. A straightforward ANN was created for each aerodynamic coefficient.

2 THE LAYOUT MATHEMATICAL REPRESENTATION

Detailed description of the MAV surface used in CFD codes is impractical for machine learning tasks. To solve this problem simplified mathematical representation was created. It is divided into 4 sub-models: wing, fuselage, tail and model of their relative location. Each sub-model approximates the surface with vector of geometrical parameters. For wing representation 8 parameters are used: area, aspect ratio, taper ratio, dihedral angle, wing setting angle, leading edge sweep angle, airfoil type identifier and airfoil thickness. Some of the parameters can be fixed by the user due to his desirable application. The range of parameters is based on existing MAVs market. Table 1 demonstrates example of parameter values for wing sub-model, which were used for testing algorithm:

To get more probable layouts taper ratio and leading edge sweep angle were chosen as functions of λ. The square S_w was set equal to 3.5 m^2 and aspect ratio was defined as

$$\lambda_w = \frac{b_w^2}{S_w}$$

taper ratio is

$$\eta(\lambda_w) = A\lambda_w^2 + B\lambda_w + C$$

and leading edge sweep angle equals

$$\chi_w(\lambda_w) = D \cdot \lambda_w - F$$

*Email addresses: vyshinsky@rambler.ru, kislovskiy@phystech.edu

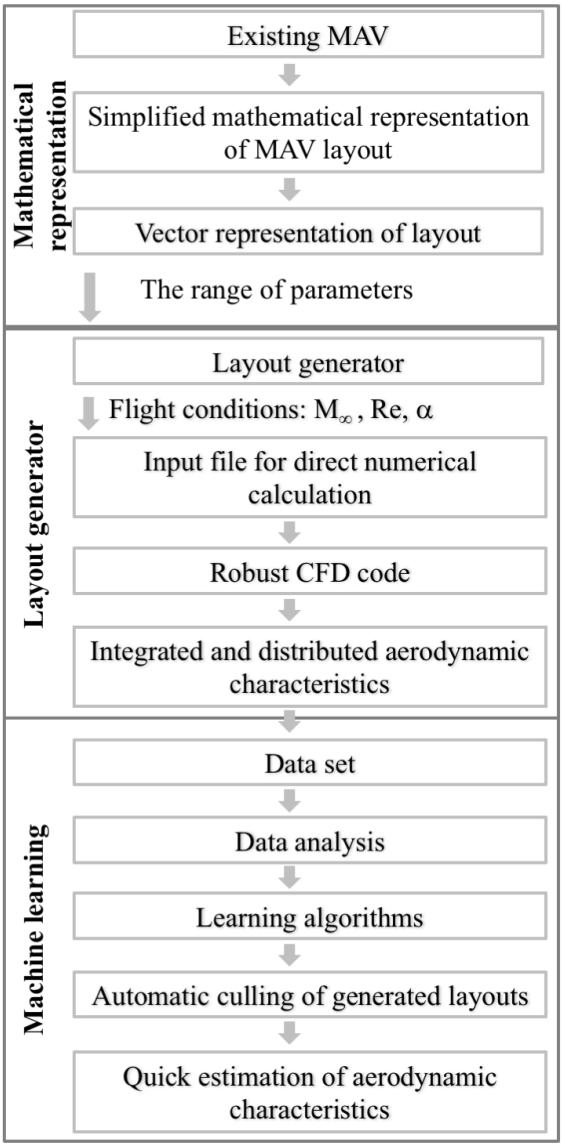


Figure 1: Key aspects of the method and logical relation of general segments

where $A = 5.33 \cdot 10^{-3}$, $B = 1.98 \cdot 10^{-1}$, $C = 2.37$, $D = 5.27 \cdot 10^{-3}$, $F = 1.9 \cdot 10^{-1}$ are empirical constants obtained from MAVs market overview.

The wing (Figure 2) is defined as follows. First wing projection on the base coordinate system is created. Then, airfoil is set in profile and 3D surface is formed. Then the wing is set on the wing setting angle. Finally, the 3D surface is changed with respect to dihedral angle Γ.

The fuselage is defined by the size of frames that separate fuselage modules (hood, nose, central and tail parts) the length and the shape of these modules. Fuselage contours functions have different analytical form for each module. Cross-section contours are described by parametrical equation

$$(\frac{2z}{h_f(x)})^2 + (\frac{2|y|}{d_f(x)})^{2+\epsilon} = 1,$$

where x is coordinate along the fuselage, $h_f(x)$ - the height of the frame, $d_f(x)$ - the width of the frame, $\epsilon \in [0, 0.5]$ - form parameter.

	Parameter	Value
1	Square	3.5 m^2
2	Aspect ratio	5 - 20
3	Taper ratio	1 - 6
4	Dihedral angle V	$0° - 4°$
5	Wing setting angle	$0° - 3°$
6	Leading edge sweep angle	$0° - 1°$
7	Airfoil type identifier	1 -551
8	Airfoil thickness	0.11 - 0.18

Table 1: Wing sub-model parameters and its limitations

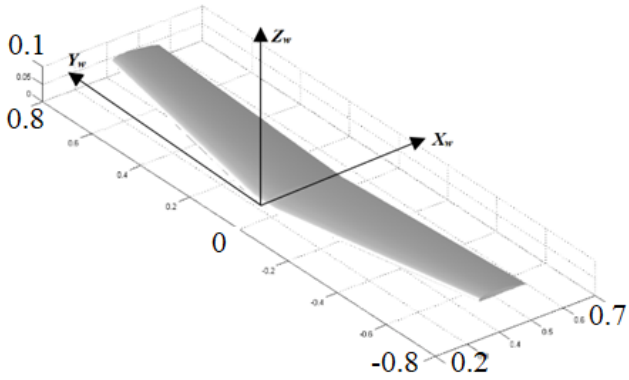


Figure 2: The general 3 dimensional view of swept tapered wing (dimension in meters)

Table 2 shows fuselage parameters applyed for its representation, where L_w is a wingspan, c_w is a root wing chord, ϕ is a wing setting angle. The numbers in value colomn describe the possible range of changes for each parameter.

Using the same algorithm of the wing definition tail unit parts are created. Vertical tail unit (Figure 4) described by the airfoil and 4 parameters given in Table 3.

As the vertical tail unit, horizontal tail unit (Figure 5) is described by airfoil and 4 parameters given in the table 4.

The relative area of the tail unit is considered relative to the wing area.

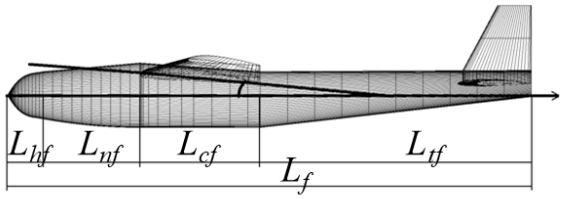


Figure 3: The side view of the fuselage with segmentations on general parts

	Parameter	Value
1	Second frame width	$(0.04 - 0.23)\frac{L_w}{2}$
2	Second frame high (H_{F2})	$(0.04 - 0.1)\frac{L_w}{2}$
3	Second frame down shift	$(0 - 0.2)H_{F2}$
4	First frame high (H_{F1})	$(0.6 - 1)H_{F2}$
5	First frame width	$(0.6 - 1)H_{F1}$
6	Middle part length (L_M)	$c_w \cos(\phi)$
7	Nose part length	$(0 - 3.5)L_M$
8	Third frame high (H_{F3})	$(0.3 - 1)H_{F2}$
9	Third frame width	$(0.3 - 1)H_{F2}$
10	Third frame down shift	$(0 - 0.2)H_{F2}$
11	Tail fuselage part length	$(0 - 4)L_M$
12	Tail fuselage part shift	$(-0.4 - 0.2)H_{F2}$
13	Fuselage back edge high (H_{tip})	$(0.1 - 0.5)H_{F2}$
14	Fuselage back edge width	$(0 - 0.5)H_{tip}$
15	Fuselage form parameter	0 - 0.5
16	Hood length	0.06

Table 2: Fuselage sub-model parameters and its limitations

Each sub-model is a variaty of points created in its own coordinate system. The sub-model of relative parts location aggregates sub-models into one base coordinate system of the MAV (Figure 6) using parallel transfer procedure for each dot.

3 Layout Generator

As far as the mathematical representation which transforms the surface (long vector) into short vector of parameters is constructed, its possible to create an algorithm wich will transform the short vector and new layouts will appear.

The layout generator randomly changes unfixed representation parameters in the limited range. Firstly, 25 layouts were used to set the constraints. The procedure above increased its number to 8659. CFD input file is created for

	Parameter	Value
1	Aria	$0.02 - 0.04$
2	Aspect ratio	$1.2 - 1.7$
3	Taper ratio	$0.3 - 0.65$
4	Airfoil thickness	$0.09 - 0.12$

Table 3: Vertical tail unit sub-model parameters and its limitations

	Parameter	Value
1	Aria	$0.03 - 0.16$
2	Aspect ratio	$3 - 5$
3	Taper ratio	$0.5 - 0.8$
4	Airfoil thickness	$0.09 - 0.12$

Table 4: Horizontal tail unit sub-model parameters and its limitations

each layout and numerical experiment is sturted in order to determine the aerodynamic characteristics. These calculations have been made using BLWF CFD-code [7] where a boundary-value problem for full velocity potential equation is solved. Viscosity is taken into account in the boundary layer approximation with fixed position of the laminar-turbulent transition. Generated layouts are presented in Figure 7. In this example all calculations were done on one flight regime. The results of computations for drag, lift coefficients and pitching moment are given in Figure 8. Each point in the graph is separate layout.

4 Artificial Neural Nets Technique for the Troblem Considered

Before the creation of the ANN the learning task should be formulated in a correct way. A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E** [8]. For quick estimation of aerodynamic characteristics **E**, **T** and **P** are defined as followes:

Task **T**: the prediction of aerodynamic coefficients;

Performance measure **P**: cost function (squared difference between the output of a neural network and calculated coefficient);

Training experience **E**: learning on dataset calculated by CFD codes.

The process of creation of the ANN consists of 8 basic steps:

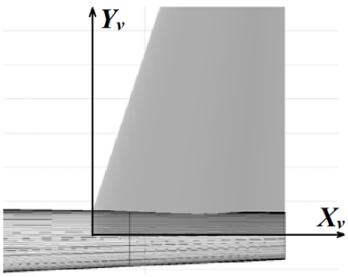


Figure 4: The side view of vertical tail unit with local coordinate system

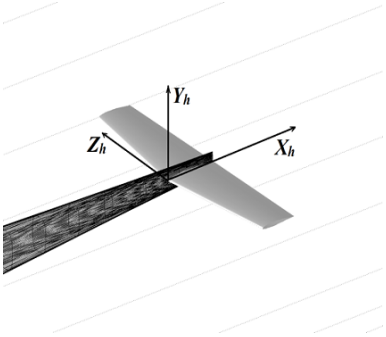


Figure 5: 3 dimensional view of the horizontal tail unit with local coordinate system

1. Collection of data for training
2. Data preparation and normalization
3. Network topology selection
4. Experimental selection of network characteristics
5. Experimental selection of training parameters
6. Training
7. Check the adequacy of training
8. Adjustment of parameters, final training

Steps 1,2 are related to data which must be presented in one vectorized form and satisfy problem statement. Both requirements were fulfilled automatically in the layout generator created with respect to it. All data were divided into 3 sets: training set - 4800 patterns (60%), validation set - 1600 patterns (20%) and test set - 1600 patterns (20%). Validation vectors are used to stop training early if the network performance on the validation vectors fails to improve or remains the same. For quick estimation of the aerodynamic characteristics direct feedforward ANN was created (step 3). The network has the following characteristics (step 4). Input layer has 100 neurons, hidden layer has 10 and output layer - 1 neuron. For each aerodynamic coefficient, separate ANN was trained. In step 5 sigmoid function was used as activation function. Neural net was trained using Lavenberg-Merquardt

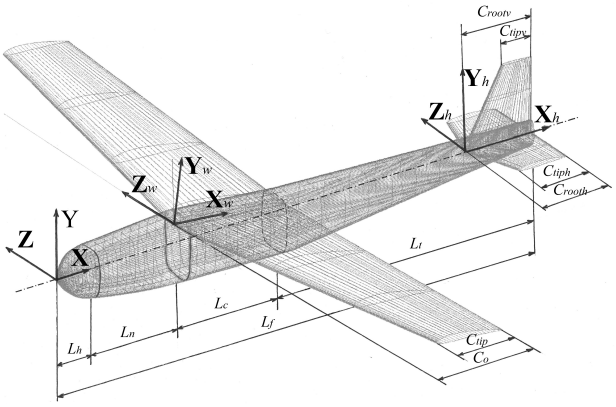


Figure 6: General view of the total MAV layout into base coordinate system

method which adaptively varies the parameter updates between the gradient descent update and the Gauss-Newton updates [9, 10]. In the gradient descent method, the sum of the squared errors is reduced by updating the parameters in the steepest-descent direction. In the Gauss-Newton method, the sum of the squared errors is reduced by assuming the least squares function is locally quadratic, and finding the minimum of the quadratic. The Levenberg-Marquardt method acts more like a gradient-descent method when the parameters are far from their optimal value, and acts more like the Gauss-Newton method when the parameters are close to their optimal value. This makes this algorithm the most widely used optimization algorithm for wide variety of problems.

The following results are given for the lift coefficient. Figure 9 demonstrates the error as a function of epoch. Training stops when network performance fails for 60 epochs in a row as shown in Figure 10. An epoch is a measure of the number of times all of the training vectors are used once to update the weights. Graphs in Figure 10 demonstrate the values of gradient, parameter mu related to Levenberg-Marquardt method and number of validation fails through the training process.

To carry out the adequacy of training 1000 new patterns were used. For each item the error was calculated which is absolute value of difference between the neural net output and the result obtained by direct calculation. The histogram shown in Figure 11 demonstrates the distribution of errors.

Performance of neural net could be estimated in a different more visible way. Figure 12 demonstrates the results of ANN lift coefficient approximation (vertical axis) versus CFD results (horizontal axis). When ANN output is equal to CFD calculation the points in the graph form a straight line.

Changing ANN parameters (step 8) it is possible to improve its performance.

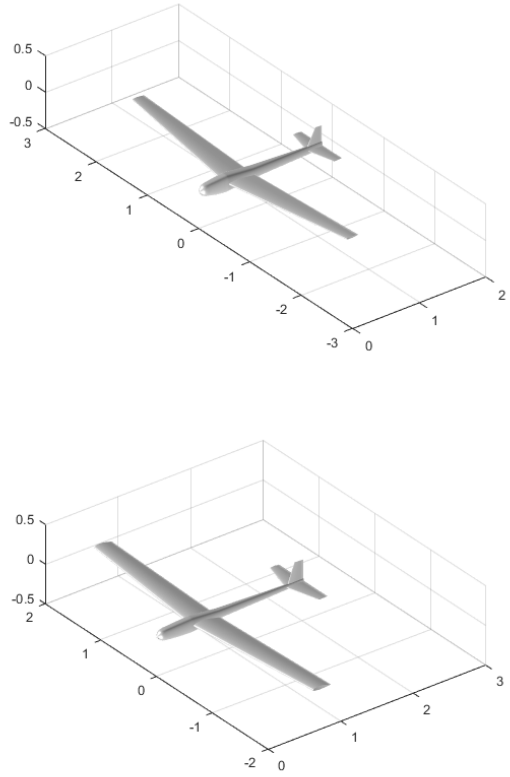


Figure 7: The examples of Generated MAV layouts. 3 dimensional view (dimensions in meters)

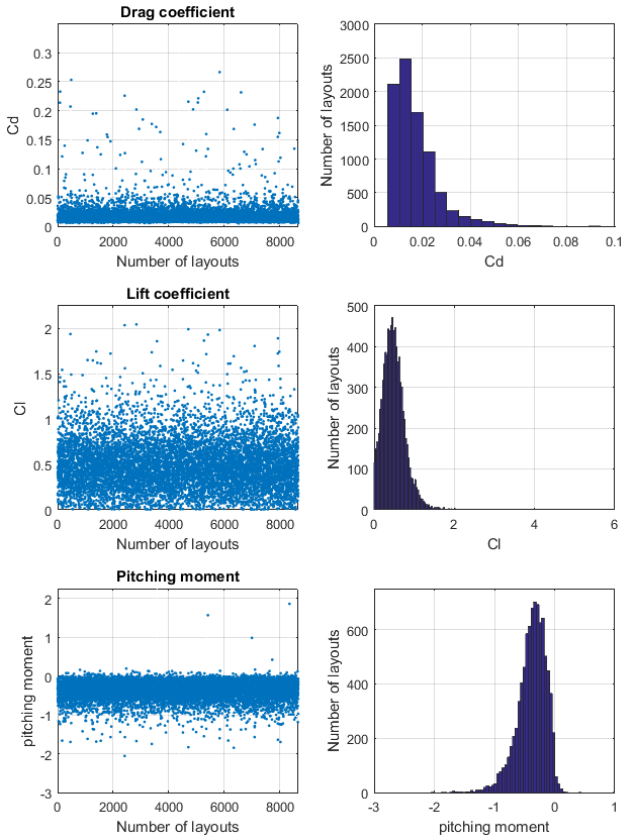


Figure 8: Drag, lift and pitching moment coefficients of the 8659 layouts generated in automation mode after culling

5 Conclusion

The algorithm developed can be applied to experimental database with real viscosity and separation effects. Proposed approach allows not only to estimate the aerodynamic coefficients of MAV but also accumulate the experience, obtained from different sources (calculations, flight and wind tunnel tests etc.), during design process. It could be implicitly reused in development of similar MAVs series. Following algorithm realization in software make it possible to solve problems of comparative analysis of the layout aerodynamic perfection and aerodynamic drag minimization.

Computational algorithms solve boundary value problems for which the correctness (existence and uniqueness of the solution, continuity dependence on boundary conditions), as a rule, is not proved. Consequently, the question of the methods accuracy remains open. In practice, the problem is solved by comparing the calculations with existing physical experiment, the data obtained with other methods and by comparison with few exact solutions. Thus, there are no rigorous general estimates of determination of integral or distributed characteristics accuracy for specific methods. Accuracy of the solutions exists only for specific types and density

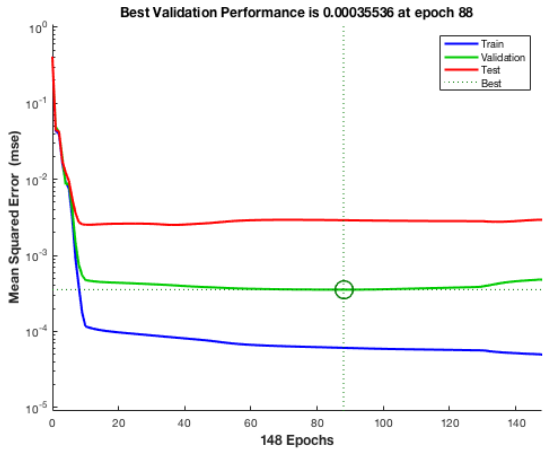


Figure 9: The process of ANN training (mean square error versus epoch)

of grids and specified parameters of the numerical scheme. In this case, the application of artificial neural networks as a universal approximator of the vehicle aerodynamic charac-

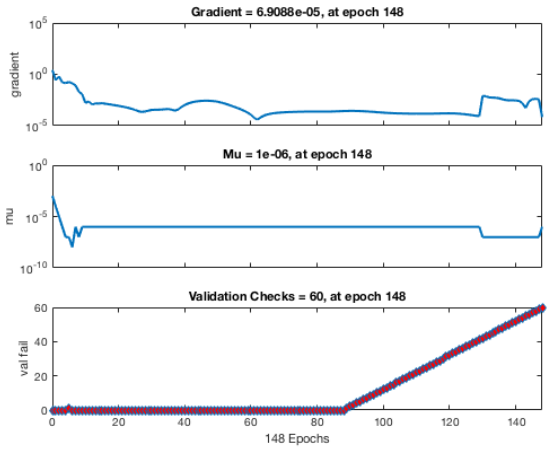


Figure 10: Training control parameters related to Levenberg-Marquardt algorithm

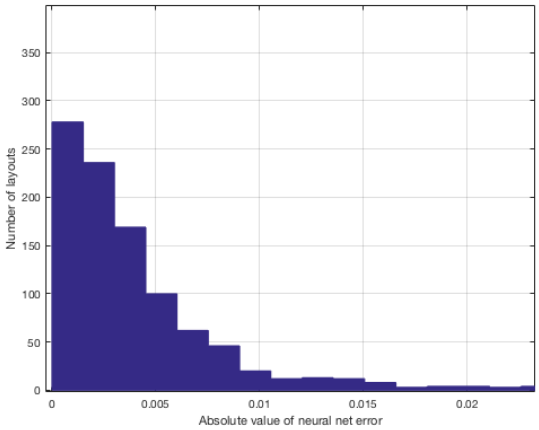


Figure 11: The error distribution on additional set (1000 items) for lift coefficient (The number of layouts versus absolute value of neural net error)

teristics provides additional advantages, mainly because it is possible to use heterogeneous data for its training.

References

[1] N. K. Madavan' M. M. Rai. Aerodynamic design using neural networks. *AIAA*, pages 1674–1691, 1998.

[2] P E. Rubbert. CFD and the changing world of airplane design. *Proceedings of the 19-th Congress of ICAS*, 1994.

[3] V.V. Vyshinsky Ye. A. Dorofeev, Yu. N. Sviridenko. CFD and the changing world of airplane design. *Proceedings of the 27th Congress of ICAS*, 2010.

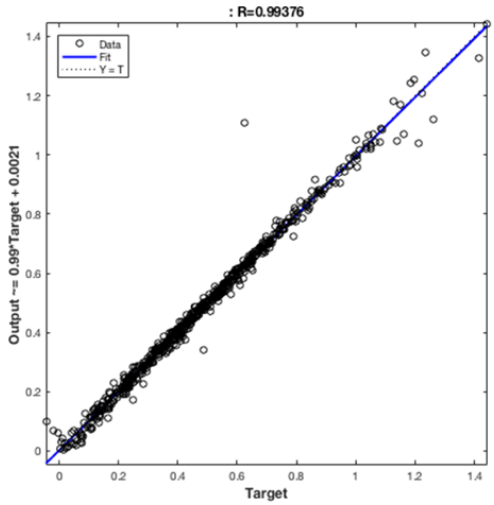


Figure 12: The ANN performance (neural net outputs versus the targets obtained from CFD calculations). If error is zero the points in the graph form a straight line.

[4] Yu.N. Sviridenko V.V. Vyshinsky A.V. Bernstein, A.P. Kouleshov. Fast aerodynamic model for design technology. *Proceedings of West-East High Speed Flow Field Conference, Moscow, Russia*, page 12, November 2007.

[5] A. O. Kislovskiy V. V. Vyshinsky. Simplified mathematical model of small sized unmanned aircraft vehicle layout. *Civil Aviation High TECHNOLOGIES*, 19(6):86–94, 2016.

[6] A. O. Kislovskiy V. V. Vyshinsky. Layout generator of small sized unmanned aerial vehicle. *Civil Aviation High TECHNOLOGIES*, 19(6):95–101, 2016.

[7] O.V. Karas V.E. Kovalev. Calcul de lecoulement transsonique autour dune configuration aile-plus- fuselage compte tenu des effects visqueux et dune region decollee mince. *La Recherche. Aerospatiale*, (1):23–38, 1994.

[8] Tom M. Mithcell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997.

[9] Henri Gavin. The levenberg-marquardt method for nonlinear least squares curve-fitting problems, 2011.

[10] A. Ranganathan. The levenberg-marquardt algorithm. *Tutoral on LM algorithm*, 11(1):101–110, 2004.

Copter Size Minimization for the IMAV-2017 Competition in Record Breaking Session

S. Serokhvostov*, and B. Makaev†
Moscow Institute of Physics and Technology
Department of Aeromechanics and Flight Engineering
140180, Gagarina street, 16, Zhukovsky, Russia

ABSTRACT

Optimization of aircraft design for the IMAV-2017 competition in Record Breaking session is investigated. Analytical research is conducted to understand the ways of optimization. A set of experimental designs was made and investigated to check the analytical results and to test technical solutions.

1 INTRODUCTION

According to the rules of IMAV-2017 competition in Record Breaking session, the goal is to lift the 0.5 kg payload for 1 minute at more than 50 cm above the ground. The winner is the aircraft with the lowest maximal dimension. Mass of the aircraft with the payload should not exceed 2 kg.

So, the optimization problem is to minimize the maximal dimension under some restrictions.

As a concept design the multi-copter was chosen. It was postulated that the maximal dimensions are defined through the number of propellers and propellers diameter. Some ways of optimization are

- to optimize the number of propellers
- to optimize the propellers number of blades
- to optimize the propeller blade shape and twist
- to optimize the diameter of propeller
- to optimize the motor
- ...

2 ANALYTICAL INVESTIGATION

To understand the influence of the abovementioned factors the mathematical model of copter was made.

The characteristic cases for the investigation were chosen as helicopter, three-copter, quad-copter and hexa-copter. If the propeller diameter is D, the maximal dimensions are

- D for 1 propeller (helicopter)
- $2D$ for 3-copter
- $(1 + 2^{0.5})D$ for 4-copter
- $3D$ for 6-copter

For the following analysis we can present this dependence as function $f(N)$. The thrust T and power P of the propeller can be expressed as

$$T = C_T \rho n^2 D^4 \tag{1}$$

$$P = C_P \rho n^3 D^5 \tag{2}$$

where C_T — thrust coefficient, C_P — power coefficient, n — rotational frequency.

If the total mass of aircraft with the payload is m and the number of propellers is N, then, neglecting the interference between the propellers

$$P = \frac{C_P}{C_T^{1.5}} \frac{\left(\frac{mg}{N}\right)^{1.5}}{\sqrt{\rho}} \frac{1}{D} \tag{3}$$

In this case, the total power of N propellers is

$$P_{sum} = \frac{C_P}{C_T^{1.5}} \frac{(mg)^{1.5}}{\sqrt{N\rho}} \frac{1}{D} \tag{4}$$

so

$$D = \frac{C_P}{C_T^{1.5}} \frac{(mg)^{1.5}}{\sqrt{N\rho}} \frac{1}{P_{sum}} \tag{5}$$

Maximal dimension MD will be

$$MD = f(N)D = \frac{f(N)}{\sqrt{N}} \frac{C_P}{C_T^{1.5}} \frac{(mg)^{1.5}}{\sqrt{\rho}} \frac{1}{P_{sum}} \tag{6}$$

If the electrical efficiency of powerplant (motor, controller, accumulator etc.) is η, then the total energy E in the accumulator required for the flight during the time t is

$$E = \frac{P_{sum} t}{\eta} \tag{7}$$

The mass of accumulator is practically proportional to the energy stored (and also depends on the maximal current of accumulator). From this, we can accept that the accumulator mass is proportional to the total power.

*Email address: serokhvostov@mail.ru
†Email address: becho15rus@gmail.com

In the first approximation we can also assume that for all the other parts of powerplant (motors, controllers etc.) the mass is proportional to the power. In this case we can say that

$$m = m_0 + \alpha P_{sum} \tag{8}$$

where α is some coefficient of proportionality, m_0 is the part of total mass that does not change with P_{max} change. Finally

$$MD = \frac{g^{1.5}}{\sqrt{\rho}} \frac{f(N)}{\sqrt{N}} \frac{C_P}{C_T^{1.5}} \frac{(m_0 + \alpha P_{sum})^{1.5}}{\alpha P_{sum}} \alpha \tag{9}$$

So, roughly, the function of MD is the multiplication of function depending on N, function depending on the propeller geometry and function depending on the powerplant characteristics. From this, one can analyze and minimize these functions separately.

First of all, for the number of propellers, the corresponding function $f(N)/N^{0.5}$ is equal to

- 1 for helicopter,
- 1.154 for 3-copter,
- 1.207 for 4-copter,
- 1.225 for 6-copter.

The best design is helicopter (1 propeller) but it was not taken into account the questions of helicopter stability. Some additional "devices" must be implemented that can increase the dimensions or mass of aircraft, so helicopter in reality is not too good. The difference between 3-copter and 4-copter is about 4.6%. But 3-copter now seems more complicated, so one of the conclusions is that the quad-copter can be good compromise for this task.

The value of $C_P/C_T^{1.5}$ (inverse figure of merit) depends on the geometry of blades and the number of blades NN. In first approximation one can assume that both C_P and C_T depend linearly on the number of blades (in reality the dependence for C_T is lower than linear and for C_P is higher than linear). In this case the value of $C_P/C_T^{1.5}$ changes as $NN^{-0.5}$. From this point of view, the more blades the better. Practically the same effect is due to the increase of blade width. But for some number of blades or some width due to the effects not taken into account there must be minimum $C_P/C_T^{1.5}$. Also, from geometry, for some number of blades and some width the blades will touch each other, and this will be geometrical limitation on the number of blades.

So, the another conclusion is to use multi-blade propellers. Also one of promising solutions is to cut the blades of larger diameter propellers.

As for the propeller blade twist, for the analysis one can use the results of [1]. For the set of small propellers one can see that it is possible to choose the propeller with the high figure of merit for different diameters.

For the third function in dependency of MD the minimum with respect to P_{max} corresponds to the condition of

$$\alpha P_{max} = 2m_0 \tag{10}$$

Corresponding value of function in this case is $3^{1.5}\alpha m_0^{0.5}/2 = 2.6\alpha m_0^{0.5}$. If we take

$$\alpha P_{max} = m_0 \tag{11}$$

then the function is $2^{1.5}\alpha m_0^{0.5}/2 = 2.83\alpha m_0^{0.5}$ The difference is about 9%. So, the optimal mass of copter for this competition can be estimated as 1–1.5 kg. One can see that it is within the limitation of 2 kg.

3 TECHNICAL PROBLEMS

From (9) the maximal dimension depends on the value of α. This parameter includes such factors as powerplant efficiency, dependency of accumulator mass on the maximal current and some other factors. So, special attention must be paid on the question of choosing the motor characteristics and accumulator characteristics.

For example, for one series of motors one can choose the different Kv. On the one hand, the motors with higher Kv seems more powerful. On the other hand, for higher Kv the reaction time is longer. This make the stabilization less effective and requires some additional power. Also, it is known that for the fixed shaft power at fixed frequency the best efficiency corresponds to the motors for which the maximal power is several times higher that required. So, the motors with the best efficiency are heavier than less effective ones. This means that lighter motor can consume more power, and the accumulator for such motor must store higher amount of energy and thus be heavier. Other thing is that higher power for the same accumulator voltage requires higher current. It is well known that accumulator with higher maximal current (for the fixed voltage and capacity) has higher mass. One can see that some compromise must be found for this situation.

Another thing that must be taken into account is the restrictions on the motor maximal frequency. Decreasing the propeller diameter for the fixed thrust leads to the frequency increasing. So, the motors with the higher working frequencies for the fixed shaft power are required. On the other hand, such a motors can be not existing in the market.

Unfortunately, now it is practically impossible to describe analytically all these peculiarities, so the optimization must be made for the discrete set of motors and accumulators.

Up to here there were no words about the frame. We assume that in the first approximation the mass of airframe is also proportional to the total power.

Another factor not taken into account is the dimensions of accumulator. The sizes of accumulator of required capacity can be comparable with the dimension of propeller. In this case it is required to make some additional frame for it.

It is evident that all these conclusions must be proved by the experiment. Some preliminary experimental investigations were made previously.

Also, some investigations are required for the designs with the propellers situating not in one plane. This enables to make lower dimension for the same propeller diameter due to the "intersection" of propellers.

Some peculiarities can occur due to the accumulator properties. First of all, for this task not only the capacity is important but also the maximal current. This is because of the fact that the accumulator must provide the required current that can be high enough. The second reason is that for the high values of current the power losses increase due to the accumulator's internal resistance. This leads to the lower accumulator efficiency. Also it is well known that the accumulator capacity depends on the discharge current. To diminish these factors one can use the accumulators with high maximal current but these accumulators have lower energy capacity.

The next problem for the accumulator is the dependence of voltage on the accumulator charge. It is well known that the fully charged LiPo accumulator has the voltage of 4.2 Volt and fully discharged accumulator has the voltage of about 3 Volt. This means that the maximal power is diminishing during the flight. So, it can be the situation that the aircraft that can stay in flight at the beginning of flight drop down after some time with some amount of energy in the accumulators.

Another factor is that copter must have some extra voltage in accumulator (comparing to the flight in perfect conditions) for the stabilization and manoeuvres.

All these reasons lead to the fact that the "real" construction will be not so optimal as the theoretical one.

4 COPTER DESIGN AND EXPERIMENTS

So, our goal was to create an aerial vehicle (AV) capable of lifting 0.5 kg cargo load to a height of at least 1 meter while staying in the air for more than 1 minute. AV geometric dimensions (maximum horizontal distance between its elements including propellers) should be minimized to an extent possible. In the above chapter we have provided the mathematical calculations associated with defining the optimal number of propellers and their blades, as well as the composition of other AV elements. We have found that the less number of propellers leads to the vehicle's smaller dimensions at the same level of thrust (subject to identical propeller characteristics for each design), so we decided to use a quadcopter as one of the most universal and sustainable designs for unmanned aerial vehicles.

At the next step we have selected powerplant and other electronic components. For this purpose we had to preliminary estimate the weight of AV, as follows:

- We estimated the battery capacity for similar purposes and this resulted with max. value 1300 mAh and 16 V (4S) (the weight of this battery type is about 170 g);
- The weight of AV carbon fiber frame was estimated to max. 80 g;
- It was decided to use a ready-made solution by installing integrated electronic speed controllers (ESC) and flight controller weighting 22 g;
- Receiver RC 10 g;
- Mounting equipment 40 g.

Taking into account 500 g of additional cargo, the total AV weight is estimated at 822 g, without motors and propellers which selection is discussed in more detail below.

One of the main methods to reduce an AV size is to use a propeller with minimal diameter taking into account allowable drop of thrust and related factors. Therefore we were selecting motors and propellers in parallel. It was decided to make the investigation "step by step", from simple case to more complex to test the solutions one by one. This gives more clear understanding of each specific factor.

We started with type BrotherHobby Tornado T1 1407 3600KV motors and 4045*3 propellers (hereinafter the first figure of a propeller model indicates the diameter in inches and the two last figures indicate its pitch). At 15 A current this type of motor with this propeller is capable of producing 535 g of thrust (data are provided by the motors manufacturers), which corresponds to total thrust of 2 kg of all 4 motors. The weight of AV equipped with these motors is 906 g. The required 60 A current is provided by the selected battery capable of generating a current up to 120 A. The estimated thrust of four motors without evaluation of the interference between the motors themselves, cargo and other parts of AV, exceeds the AV weight more than two times.

The experiment was carried out.

A box suspended under the AV (Figure 1) was selected as the cargo (the box larger face was positioned horizontally). AV took off at 90% of motors load, which reached 95% in stable flight, after 35 s of flight one motor wiring could not stand the power supply and AV fell down because of the motor burn-out.

When the cargo position was changed (the box smaller face was positioned along the air flow from propellers), AV could hardly manage to stay in the air for 1 minute required. (Figure 2). These experiments have helped us to find out that one of the main factors effecting AV performance with cargo is the cargo position (cargo's the smallest face should be positioned perpendicular to the air flow produced by propellers to reduce the aerodynamic force on cargo surface). Besides we considered 1407 3600 KV motors to be not powerful enough for the purpose. So, one of the conclusions is that the size and shape of cargo will have a major impact on AV power plant performance, so the AV frame should be designed for a specific cargo. Next, we considered two options for resolving this problem:



Figure 1: Copter of first design in flight with cargo fixed "horizontally"



Figure 2: Copter of first design in flight with cargo fixed "vertically"

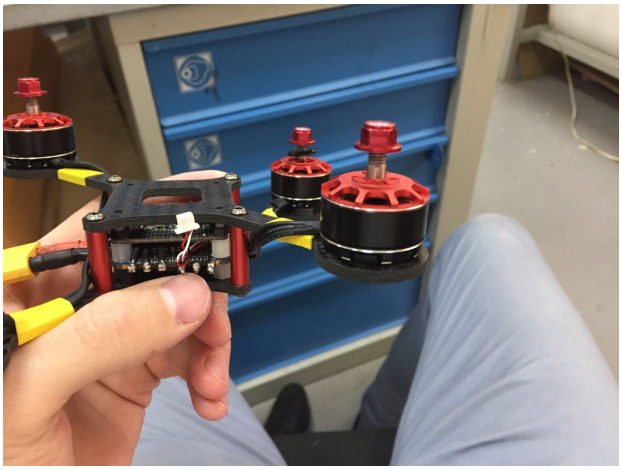


Figure 3: Double deck frame

- Extension of the quadcopter main diagonal to divert the air flow produced by propellers from the cargo surface;
- Use of more powerful motors and, accordingly, larger propellers.

To select a more efficient option we have conducted the experiment as follows: by using motors of Titan TS2307-2300KV type with 5046*3 propellers (of higher diameter comparing to the first design) we reduced the frame diagonal keeping the AV size unchanged (as it was in the first experiment).

We carried out a similar flight with the same cargo fixed along the air flow. The AV robustly stayed in the air for 1 minute at 65% of motors load.

Thus, we have found that for this specific shape of cargo the most efficient solution is to use more powerful motors and bigger propellers, despite the increase of interference (overlapping) effect, and this finding is proved by the 1 minute stable flight of AV, which had the same size.

Having selected 2307 2300KV motors and 5046*3 propellers we considered the two following ways to make the AV size smaller:

- To reduce the propellers diameter by partial cutting the blades of existing ones;
- To minimize the frame size.

To minimize the frame size a double deck frame was designed (Figure 3). In this case the AV minimum size can theoretically be equal to $2D$, where D is the propeller diameter. In our case, when using 5045 propellers the AV size can be equal to 25.4 cm (10 inches), which is optimal for the design of double deck frame and these propellers.

At the time of these experiments the size and shape of cargo to be lifted by the AVs of the competition participants were published on the IMAV web-site. The width of the cargo



Figure 4: Copter with cargo on threads

main part is 180 cm, the length is 580 cm. As its weight is 500 g, the cargo is strongly subjected to the wind and thus heavily spoils the aerodynamics of AV + cargo system.

We have made the model of this cargo at the scale of 1:1 to investigate the influence of its shape and inertia.

When the cargo was fixed under the bottom of our AV, it was not possible to lift it because of the strong counter force from the propellers air flow (there was an appropriate experiment). We may say with confidence that our system of AV with the cargo of this size and shape can not operate efficiently, if the cargo is fixed under the AV bottom.

We considered the following ways to solve this problem:

- Suspension of the cargo on threads to reduce the force acting on it;
- Rigid fastening of cargo on a long support underneath AV;
- Rigid fastening of cargo above the AV, where the impact of air flow force produced by the propellers is much less.

When the cargo was fixed with a 30cm-long cable (see Figure 4) the most of the air flow still affected the cargo. Under these conditions the AV took off at 80% of motors load, and the entire system fluctuations appeared immediately making the AV control extremely difficult and unsafe. Then we have tried to extend the cable, assuming that it would result in

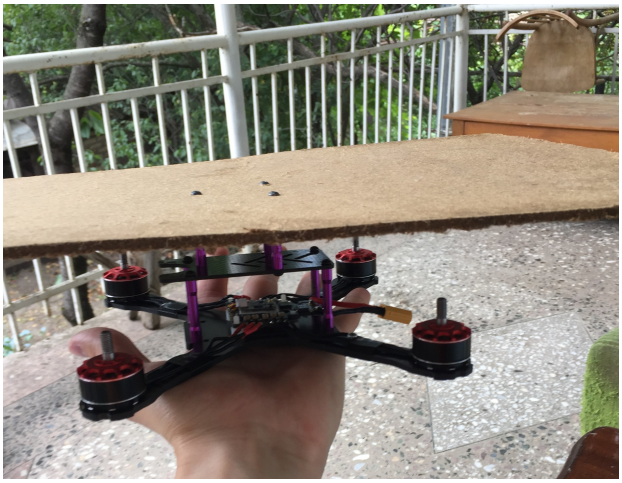


Figure 5: ”Cargo over AV” first design

reducing the airflow impact and the fluctuations of the system would become acceptable for the safe AV control. However even using a cable of 1.5 m length we did not achieve the desired result.

The last option for fixing cargo is not applicable for our system, as there is no place on the AV to mount the fasteners for the cargo.

But to test the idea of ”cargo over AV” we assembled a quadcopter with a frame diagonal 210 cm long and fixed a board with dimensions declared for the cargo to be used in the competition. We did not overload the AV in the first experiment, therefore the weight of the board was 250 g instead of 0.5 kg (see Figure 5). In the first case the distance between the board and propellers plane was 5 cm. The AV took off at 65% of motors load. Then we increased the distance to 10 cm (Figure 6), and the take-off was operated at 50% of the motors load. It can be said with confidence, that when cargo, which size is similar to that of AV, is fixed over the AV, such operation is much more efficient than with cargo fixed beneath the AV. However ”cargo over AV” operation has its shortfalls too: the system centre of mass shifts upward thus affecting the stability of AV in the flight.

When we had completed the motors selection and the cargo loading and fixation scheme, we started designing the AV that meets our requirements.

The task of optimizing the AV size in this case becomes more complex and from our point of view may be divided into 2 parts:

1. Layout of all 4 motors at the same height level. Then the minimum size of AV that can be achieved is $2.4D$. This leaves a square form gap which can be used to install the battery below the propellers plane, which in turn contributes to lowering the centre of mass and increasing the AV flight stability.



Figure 6: "Cargo over AV" second design

2. Layout of the motors at different levels (using double deck frame), then the minimum size of AV will be $2D$, as already mentioned above. However, the implementation of this scheme is somewhat more complicated. If the distance between motors reduced, AV flight stability becomes more difficult to be achieved due to decreasing of relative force moments. This may lead to the burn-out of motors, which will operate at 80-85% load level with heavy interference and the specified cargo. In this case it will also be impossible to lower the battery installation for the appropriate lowering AV centre of mass.

As the first step we have chosen the first type (Figure 7). Then, for the realization of this the new frame was designed. The drives were fixed in one plane, accumulator was placed below the propellers' plane. Also the special platform was made for the cargo fixing. Experiment was conducted with the cargo with the form and the dimensions required. The flight was stable and the motors' power was less than 60% of maximal value. As the results were very good it was decided to realize the second type. For this it was not necessary to make double-deck frame, we have used spacers to make two diagonal motors 4mm lower than others. This was enough to mount the motors as close as possible (Figure 8). In this case the final maximal AV dimension has become 26.8 sm (10.6 inches), see Figure 9. The test flight has been conducted, and the aircraft flight time was 1 min.

Some parameters of the final design are

Accumulator mass — 170g

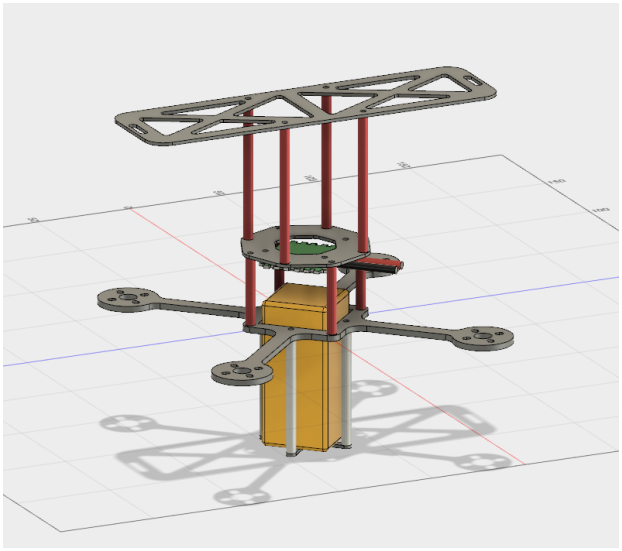


Figure 7: Airframe design for the cargo above the copter.

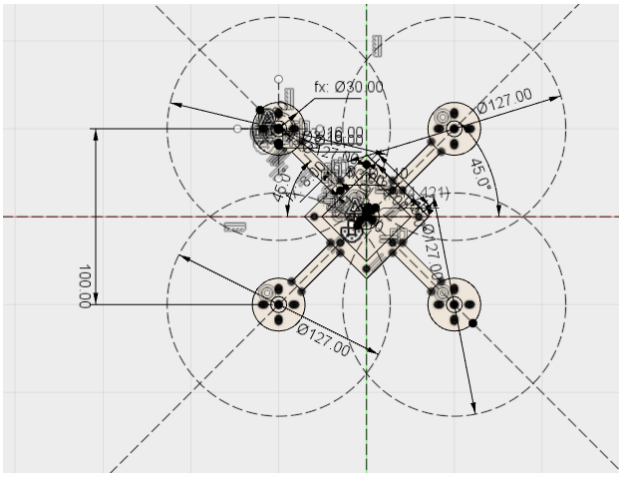


Figure 8: Final airframe design



Figure 9: Final design in flight

Total mass (without cargo) — 456 g

One can see that the value of mass is about the one obtained from theoretical investigation.

The charge in the accumulator after the 1 minute of flight was about a half of total accumulator capacity (650 mAh). This enables to estimate the current and power from accumulator, mean current is about 39 A (about 10 A per 1 motor), mean power is about 624 Watt.

5 Conclusion

1. Analytical investigations for the problem considered were conducted and it was found that the rational total mass is one-two masses of cargo; multi-blade propellers must be used, the optimal number of propellers is one and quadcopter gives the maximal dimension of 20% higher than for one propeller design.

2. A set of designs was made to check the analytical results and find good technical solutions. For the task investigated and components available the main solutions are: the optimal place of the cargo is above the copter; the maximal dimension of $2D$ can be made by placing the propellers in two parallel planes.

3. Copter mass for the final design coincides with the estimated one of theoretical investigation.

References

[1] R. Deters and M. Selig. Static testing of micro propellers. In *26th AIAA Applied Aerodynamics Conference*, 2008.

Bond Graph based design tool for a passive rotation flapping wing

Le Anh Doan*, Christophe Delebarre, Sebastien Grondel, Eric Cattan

Abstract— Micro air vehicles (MAVs) are becoming more popular over larger unmanned aerial vehicles (UAVs) as they are easily portable, more discreet and less dangerous in case of a crash. Among common types of MAVs, the flapping wing configuration shows incredibly potential flying skills such as hovering, flying backward and recovering after shock. However, designer faced many difficulties due to the micro-size of the MAVs. In this work, we propose a numerical model based on Bond Graph for our flapping MAV which allows us to analyze the wing kinematic and thus predict the total lift force. This Bond Graph model employs the quasi-steady aerodynamic theory and the Lagrange dynamic equation as the main principles. A proper wing kinematic which allows the enhancement of the total lift force could be derived from this model based on the non-linear optimization of the system's sensitivity parameters such as spring stiffness, working frequency and input voltage... A prototype is fabricated and characterized. Comparing the experiment and the simulation, the model is able to predict the wing movement and the mean lift force, and therefore could be used as a design tool. A take-off demonstration is provided to confirm our results.

***Keywords*—**UAV, MAV, flapping wing, numerical model, Bond Graph.

1 INTRODUCTION

The performance of the existing designs of the flapping wing MAVs are worse compared to the fixed and rotary wing groups. Low Reynolds number condition leads to highly unsteady aerodynamics of such vehicles). Nevertheless, the perspective of potentially achieving the exceptional flying performances has prompted a significant amount of research on the kinematics and aerodynamics of flapping flight in nature [1], [2], [3]. Subsequently, several studies have considered how vehicle designs could mimic the function or the form of flying organisms [4], [5], [6]. Here we introduce three outstanding existing insect-like flapping MAVs including DelFly, AV Hummingbird and Robobee. The DelFly is a tail fully controllable MAV [7]. Its Micro version is currently accepted as the smallest free flying controllable flapping wing MAV equipped with a camera and a video transmitter. This 10 cm wing span vehicle weighs 3.07 grams and can fly around 3 minutes. The 16 cm span and 19 g AV Hummingbird is a tailless remote controlled NAV built to mimic a hummingbird [8]. The tailless design makes it closer to a real flying humming bird, but leads to a passively unstable attitude. As a result, it needs a more advanced control system to stabilize the vehicle. Equipped with a small video camera for the purpose of surveillance and reconnaissance, this vehicle operates in the air for up to 11 minutes at 18km/h. Inspired by the biology of a bee, the Harvard RoboBee [9] is the smallest and lightest MAV that can perform controlled hovering. This vehicle weighs less than one-tenth of a gram, and flies using "artificial muscles" composed of materials that contract when a voltage is applied.

As the first example of MAV is passively stable with tail and the last one is limited in payload, it is thus decided to develop a flapping MAV mimicking the hummingbird. The developed MAV, however, is lighter in weight compared to the AV Hummingbird. To preserve the wing kinetic energy and to achieve a capable resonant system, we directly drive our flapping wing MAV using conventional DC motors coupled with helical springs. Flexible part that is added to the wing contributes to the wing's passive rotational movement. With the presence of elastic components in the system, higher efficiency would be achieved. The concept of this flapping MAV is shown in the Figure 1. This configuration allows us to utilize some of the off-the-shelf parts as well as available technology framework.

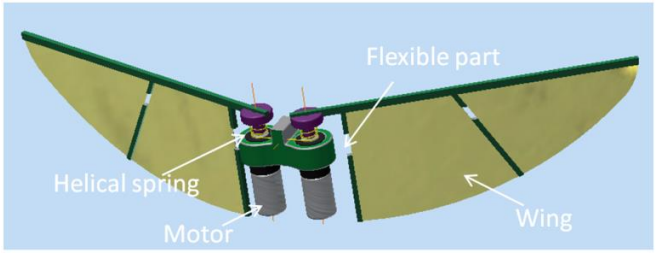


Figure 1: MAV concept

Since the studied system combines the mechanical, the electromagnetic and the aerodynamics fields, a unique model Bond Graph formalism has been set up. This kind of formalism is widely used to model multiphysics systems and energetics efficiency. Combining the Lagrange dynamic equations with a quasi-steady model of aerodynamic forces, this model serves as a tool for diagnostic system's performance. Thanks to the simulation, the MAV is able to operate at its resonant frequency with a specific torsion spring. Further optimization on the stiffness of flexible part contributes to a proper phase shift between flapping and rotational movements, which certainly results in an enhancement of total lift force. To prove the rightness of the model, a prototype is fabricated with components whose characteristics are retrieved from the simulation. The wing's motion tracked with a high-speed camera and the mean lift force measurement are used to validate the model. A take-off demonstration is also provided to confirm our results.

This article is inspired by the work of L. Hines et. al in [10] with three main different contributions. Firstly, the flexible part made by a piece of rubber is positioned along the longest chord of the wing. This helps reducing the length of the wing offset which brings the wing and also the center of

Le Anh Doan is with the University of Valenciennes, IEMN, CNRS UMR-8520, Valenciennes, France (e-mail: Leanh.Doan@etu-valenciennes.fr).

the lift closer to the actuator. This configuration also helps to vary the stiffness of the flexible part easily by modifying the rubber pieces 'dimensions. Secondly, simulation part shows the effort to approach the resonance to maximum lift forces by choosing a proper torsional spring. Lastly, to fulfill a design tool, the effect of wing offset d_w and wing flexure stiffness K_w on rotational amplitude, lift force and phase shift is also taken into account.

2 MAV MODEL

2.1 The Word Bond Graph of the MAV

The Word Bond Graph in Figure 2 reveals the main components of MAV dynamic model. The model is similar to the well-known block diagram, with the major difference that the "bonds" which link the elements together represent bi-directional exchange of physical power. Each bond depicts instantaneous flow of energy or power denoted by a pair of power variables called flow and effort. Flapping motion of wing is induced with a sinusoidal input voltage to each motor $v = Asin(2\pi ft)$ where A is the peak-to-peak voltage, f is the operating frequency. The bond next to the wave generator block would present the flow of electrical power and the power variables would be the current (i) and the voltage (v), whose product is power (P_{in}). Likewise, the motor output angular velocity (ω_l) and the torque (τ_l) are flow and effort of the corresponding power ($P_{mechanic}$). A quasi-steady model is employed to model the aerodynamic forces on our passively rotating flapping wing with the aim to predict wing motion (flapping angle θ and rotational angle φ) and lift.

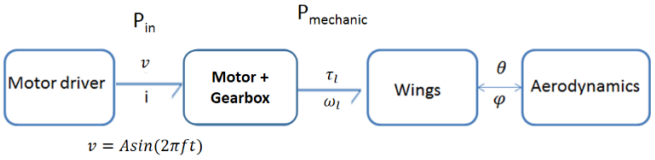


Figure 2: The Word Bond Graph of the MAV

2.2 Motor driver and geared motor model

While motor driver can be easily presented by a modulated effort source *MSe*, a model for a DC motor connected through a gear reduction is a little more complicated as shown in Figure 3.

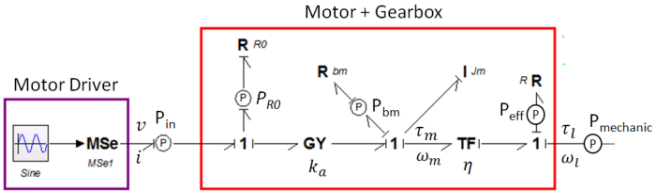


Figure 3: Motor driver and motor with gearbox blocks

The mature armature wingdings are introduced by the R_0 element. A gyrator GY with the armature constant, k_a, converts input (v) voltage into angular velocity (ω_m). A transformer element TF stands for the gearbox with the gears ratio of η. J_m and b_m are the rotor inertia and motor rotational damping values respectively. The R element right next to the TF element demonstrates the gearbox efficiency. Some power sensors are integrated in the Bond Graph model to inspect input power (P_{in}), dissipated power (P_{R0}, P_{bm} *and* P_{eff}) and the power supplied to the wing ($P_{mechanic}$).

2.3 Aerodynamic model

In the absence of skin friction, to model the aerodynamic forces on thin flapping wings, we employ a quasi-steady model [11], where the instantaneous aerodynamic forces on the wing are approximated using Blade Element Method. In this method, each wing is divided into a set of cross-section strips, each of width *dr*, and at a mean radius *r* from the axis of flapping. The instantaneous forces may be represented as the sum of three components, each acting normal to the wing surface:

$$F_{Lift} = F_{trans} + F_{rot} + F_{air} \quad (1)$$

where F_{trans} is the translational force, F_{rot} is the rotational force. The added air mass force F_{air} is the inertia of the airflow generated by unsteady wing motion.

2.4 Wings model

The dynamic model of the wing is formulated in Lagrangian form with two coordinates of flapping (θ) and rotational angle (φ).

$$L = T - V = \frac{1}{2}m_w\vec{v}.\vec{v} + \frac{1}{2}J_w\vec{\omega}.\vec{\omega} - \frac{1}{2}K_w\varphi^2 - \frac{1}{2}K_s\theta^2 \quad (2)$$

where T is the total kinetic energy of the system, equalizing the sum of the kinetic energies of the wing translation and rotation, and V is the potential energy of the system caused by the torsion spring and the flexible part with corresponding stiffness of K_s and K_w. m_w is the wing mass and J_w is the wing inertia. The motion equations of the wing could be drawn from the derivation of the Lagrangian equations.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\varphi}}\right) - \frac{\partial L}{\partial \varphi} = \vec{M}_{aero\_rot} - b_w\dot{\varphi} \quad (3)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \vec{M}_{drive} + \vec{M}_{aero\_flap} \quad (4)$$

where the $\vec{M}_{drive} = \tau_l$ is the driving flapping torque and also the torque provided by the motor. b_w is the damping of flexible part. $\vec{M}_{aero\_rot}$ and $\vec{M}_{aero\_flap}$ are the moments due to the aerodynamic forces in the rotational and flapping direction. This approach for passive wing is inspired by the work of [10]. Reader should refer to this work for full derivation of dynamic equations.

It is noted that Equations 3 and 4 have the second order form of a spring-mass-damper system with nonlinear coefficients. They can be presented in Bond Graph by two different set of C, I, R elements for each dynamic coordinate as in Figure 4. Apart from the aerodynamic moment, the two above MSE elements also include moments caused by the gyroscopic effect and centrifugal force acted on flapping and rotational axes. Wing's Bond Graph elements can be found in Appendix To form the MAV's model, all of the sub-systems are linked together through the common bonds, presenting the power transfer as shown in Figure 5.

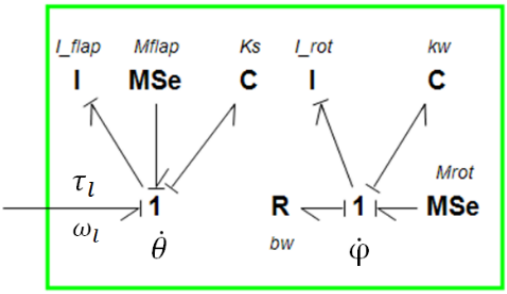


Figure 4: MAV's wing block

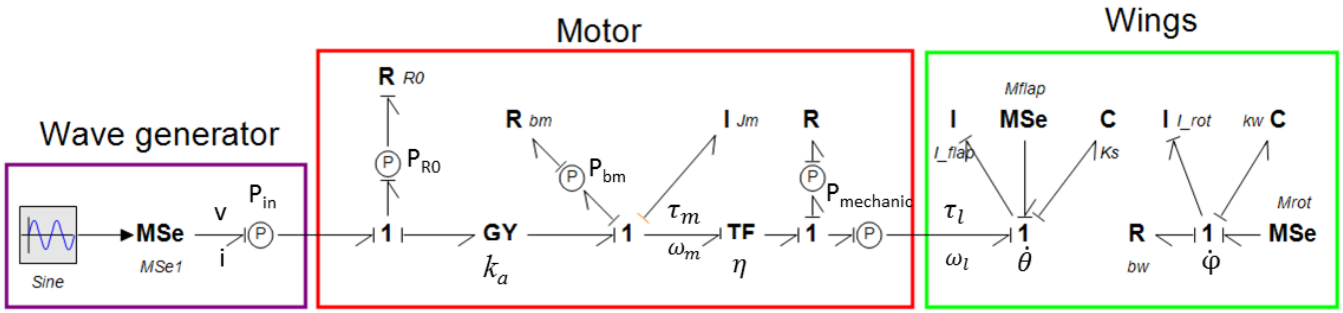


Figure 5: Bond Graph presentation of MAV

2.6 *Vertical take-off model*

In order for a fly vehicle to rise into air, the lift force (F_{Lift}) created must be at least greater than or equal to the force of gravity ($F_{Gravity}$) as shown in (5):

$$m_{MAV}\ddot{z} + f_{viscous} = F_{Lift} - F_{Gravity} \quad (5)$$

where z is the prototype's altitude and m_{MAV} is the mass of prototype. Suppose that the prototype moves at relatively low speeds through the air, the viscous resistance $f_{viscous}$ is approximately proportional to its velocity:

$$f_{viscous} = -b_{vicous}\,\dot{z} \quad (6)$$

where $b_{viscous}$ is the viscous coefficient. A 1-junction is employed to present Equation 5 as in the vertical take-off block.

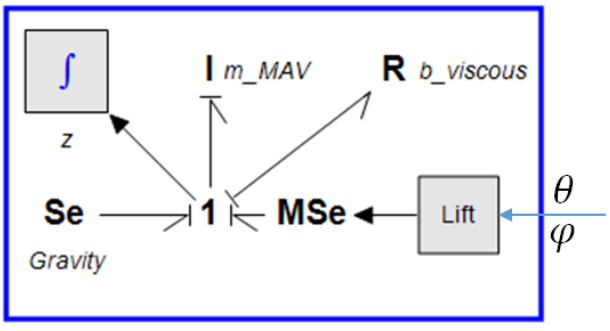


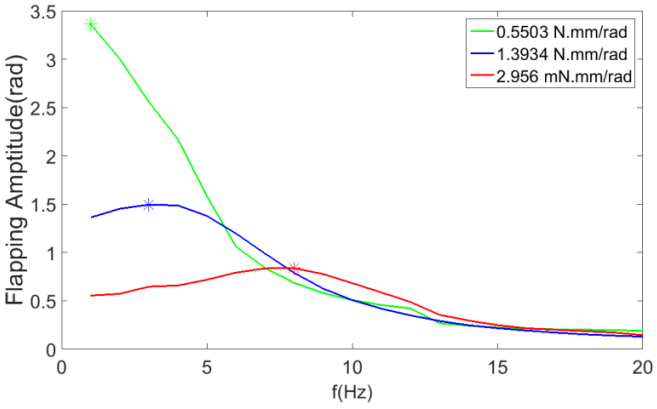
Figure 6: Vertical take-off block

3 SIMULATION

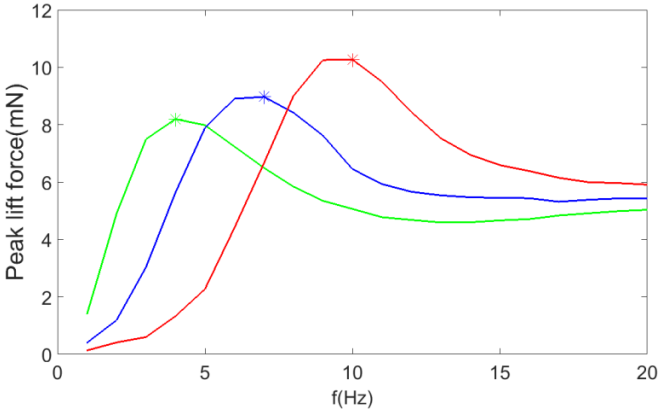
3.1 *Sensitivity to spring stiffness and driving frequency*

Three stiffness values of springs are tested on a range of input frequency from 1 to 20 Hz. The system input voltage isheld at 2V to limit the high flapping angle of the least stiff elastic element. Different colors are used to specify different elastic element stiffness systems as shown in following figures. While the system with the highest spring stiffness resonates at 8 Hz, resonance of the lowest one could not be determined. Maximum flapping amplitude of the remaining system occurs at 3 Hz.

Peak lift force, however, happens beyond the flapping resonance. At low flapping frequencies, the wing speed is too low to produce remarkable lift or wing rotation. In ascending order of stiffness, maximum mean lift occurs at 4 Hz, 7 Hz and 10 Hz respectively. It is noted that, the highest spring's stiffness brought the peak of lift close to the frequency at which the resonance happens. Performing this system at its resonance frequency allows us to optimize the system efficiency without much decreasing in lift. A compromise between wing speed and flapping angle amplitude is always the key-point to enhance the system performance.



(a)

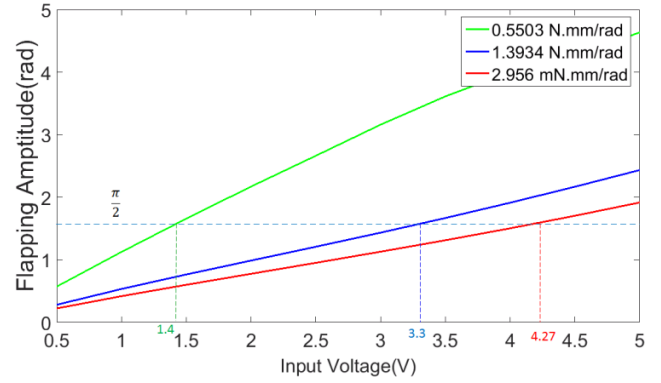


(b)

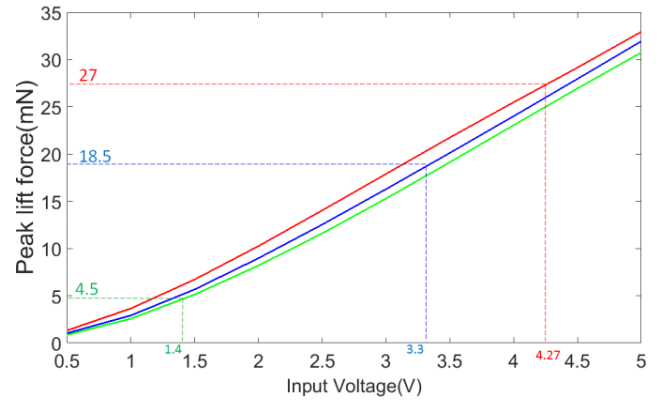
Figure 7: Effect of spring stiffness K_s and flapping frequency on flapping amplitude (a) and lift force (b)

3.2 *Sensitivity to input voltage*

A sweep of input voltage amplitudes is simulated; the results can be seen in the next figure. Depending on the spring stiffness, the driving frequency is set to the corresponding value where the highest lift value was observed as in the previous part.



(a)



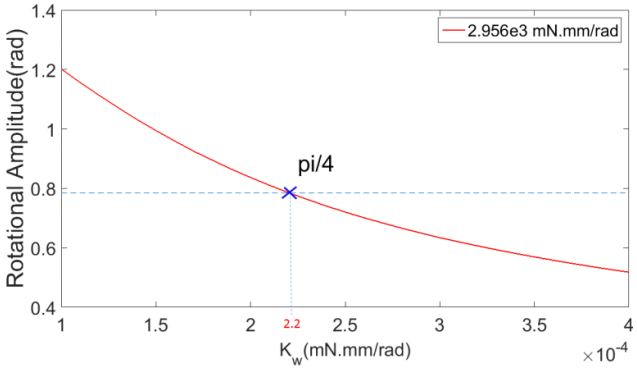
(b)

Figure 8: Effect of spring stiffness K_s and input voltage on flapping amplitude (a) and lift force (b)

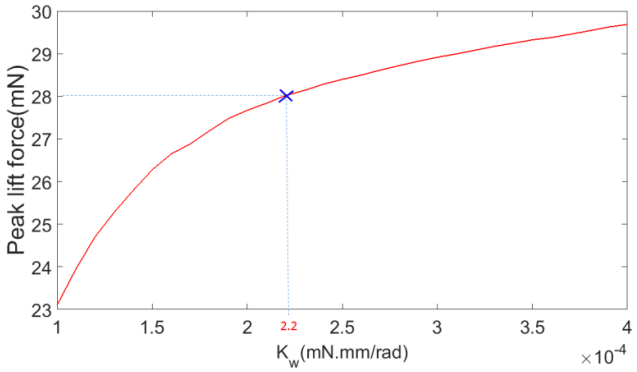
Figure 8(a) illustrates well a nearly linear relationship between flapping amplitude and the input voltage in the range of examination. Flapping angle is inversely proportional to the value of spring's stiffness. In reality, the flapping amplitude should not exceed pi/2 to avoid wings 'collision. With this constraint, the reachable lift forces at this flapping amplitude are 4.5 mN, 18.5 mN and 27 mN as shown in Figure 8(b). In conclusion, a system with maximum stiffness (2.956e3 mN.mm/rad) activating at 4.27 V and 10 Hz will be our ultimate solution.

3.3 *Sensitivity to wing flexure stiffness*

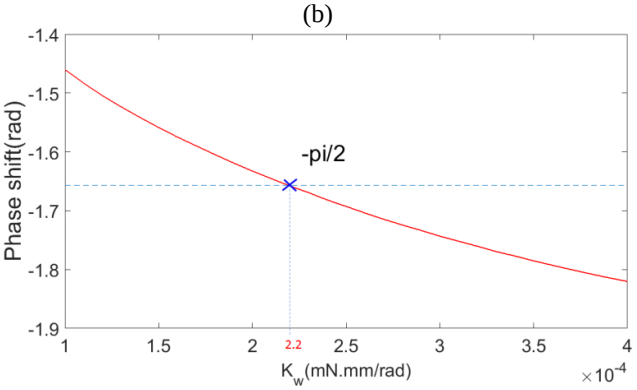
The system with spring stiffness of 2.956e3 mN.mm/rad isstimulated by an input voltage of *4.27sin(2π10t)*. Some interesting discoveries could be seen in Figure 9. It is better to remind that phase shift is the difference between rotational angle and flapping angle, a negative phase shift means the former angle is lag behind the latter and vice versa.



(a)



(b)



(c)

Figure 9: Effect of wing flexure stiffness K_w on rotational amplitude (a) lift force (b) and phase shift (c)

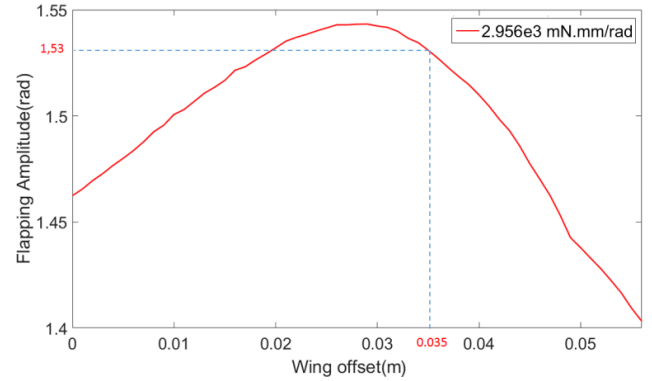
It is obvious that larger value of wing flexure stiffness results in smaller rotational amplitude. A lower value than the minimum bound of inspection range (1e-4 N.m/rad) causes over rotation and the wing exhibits erratic and unsteady behavior. Lower driving frequency could generate an

acceptable wing motion, however it will not be the goal of this scope. Raising the stiffness does slowly increase the total lift force, on the other hand it also augments the angle of attack. If this angle exceeds $\pi/4$, drag coefficient will play a major role while lift coefficient gradually degrades. This means thatthe total force is less directed vertically but more horizontally.

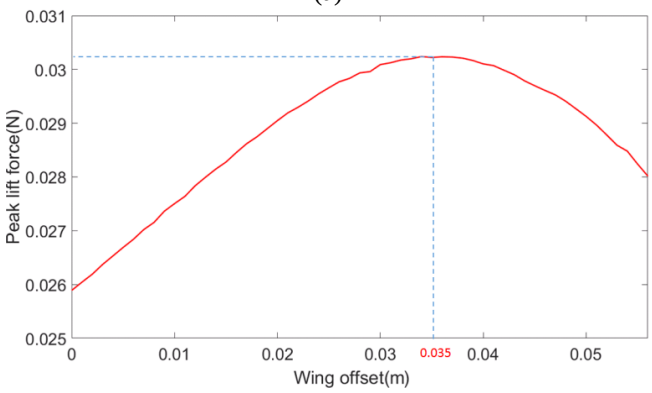
We always expect to achieve a phase shift close to $-\pi/2$ because it leads to the highest rotational angle which occurs at midpoint of flapping trajectory where the maximum flapping velocity generates an ultimate total lift force. Fortunately, all of our desires are achieved with this configuration. This system has $\pi/4$ of rotation angle at mid-stroke and $-\pi/2$ of phase shift at a flexure stiffness of 2.2e-4 N.m/rad.

3.4 *Sensitivity to wing offset*

In this part, we use the same setting as the previous part except that the variable is now the wing offset.



(a)



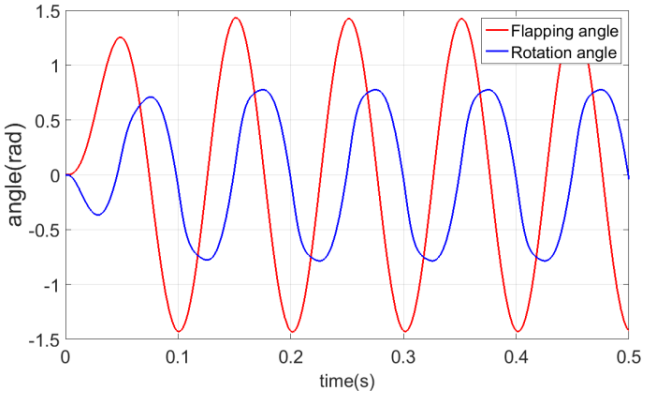
(b)

Figure 10: Effect of wing offset d_w on Flapping amplitude (a) and lift force (b).

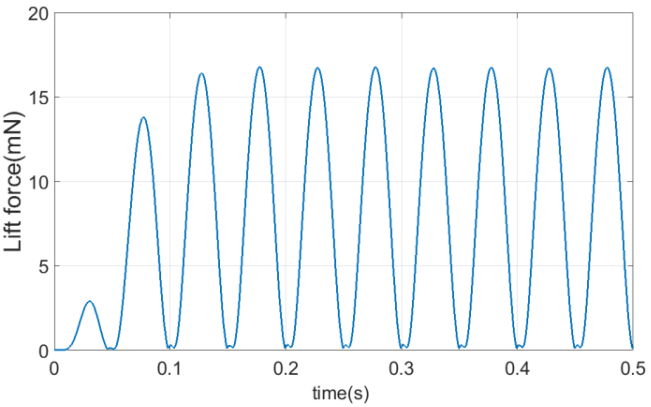
It is apparent that the wing translational velocity at the mid-chord of each wing strip is proportional to wing offset, so increasing the wing offset improves the translational force. This is well demonstrated in the range of 0 to 35 mm. However, increasing the wing offset also augments the moment inertia about the flapping axis and therefore reduces the flapping movement. While the wing size is kept constant, greater wing offset also effectively increases the damping force which can be limited given maximum motor torque. As can be seen in the Figure 10, when the wing offset is beyond 35 mm, the lift force starts to reduce. Another disadvantage of raising this parameter is that it moves the center of lift far from center gravity and makes the prototype more difficult to stabilize. Bigger size of the prototype is also another drawback. A table summarizes the optimized parameters can be found in Appendix.

3.5 *Dynamic simulation results*

After running the simulation with all the parameters found from previous discussions, a satisfactory wing's motion is portrayed in Figure 11. The rotation trajectory is $\pi/4$ and lags behind the flapping by nearly $\pi/2$. Notably, the force shown in Figure 11(a) is the cycle-averaged lift which is in the vertical body direction. The peak lift with the force of 17 mN occurs at each mid-stroke.



(a)



(b)

Figure 11: Dynamic simulation of a proper wing trajectory (a) and resulting lift force (b)

The results from the added vertical take-off block are illustrated in the following picture. The MAV reaches the altitude of 6 cm after 0.5 s with a 100mA peak sinusoidal input current.

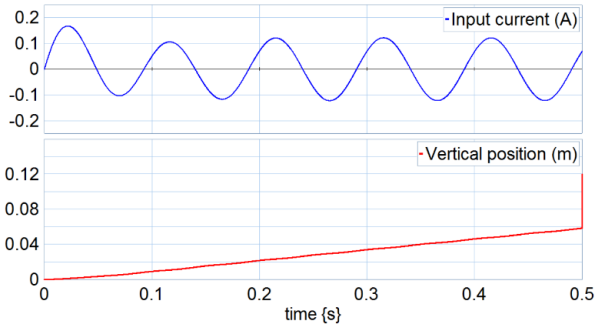


Figure 12: Simulation of vertical take-off

4 EXPERIMENT

After building the MAV model, we can now validate whether it reproduces the system behavior within acceptable bounds with the recommended parameters drawn from the optimization step. This section starts by the preparation of main components and materials for the MAV fabrication. Then a set of validation experiments including wings movement observation and total mean lift force estimation are conducted to confirm the rightness of our model.

4.1 Component fabrication

A tiny DC pager-motor named GM15A with a planetary gearhead 25:1 reduction has been chosen as the main actuator. It is designed for a 3V nominal operation, giving 920RPM drawing 100mA. The motor is controlled by a motor driver named Pololu DRV8835. Our flapping system is induced with a Pulse Width Modulation (PWM) approximated sinusoidal voltage generated by our motor driver. The wing should be as light as possible but not flexible because it is supposed to be rigid except at the flexible part and near the rotational axis.

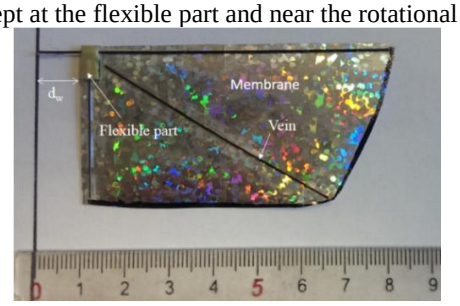


Figure 13: Fabricated Wing

The wing has the length of 8.5 cm and the maximum chord length is 3.5cm. The flexible part is made by a piece of rubber and its rotational stiffness depends on its length. d_w is the offset distance from the motor output shaft to the free end of leading edge. The membrane is made by very thin film with the thickness of $25\mu m$. The mass of the wing is 0.12 g measured by a microbalance Mettler Toledo.

4.2 Wing kinematic observation

To observe the wing movement, a high speed camera has been positioned in front of the wing. The half prototype is activated by an input voltage of $4.5\sin(2\pi 10t)$ (V). A $\pi/4$ of rotational angle at the mid-stroke is the validation for the mathematical model

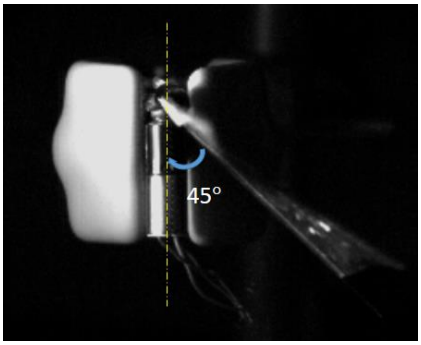


Figure 14: Diagram of wing observation experiment set-up

4.3 Mean lift force measurement

The wings with the appropriate kinematic were employed in the lift force measurement experiment. The whole set-up could be seen in the Figure 15.

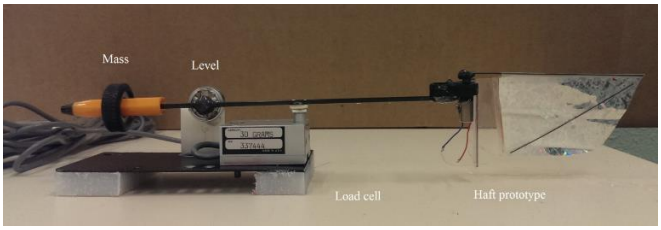


Figure 15: Lift measurement set-up

The moment created by the weight of the half prototype and the "Mass" have the same value but in the reverse direction counting at the center rotation of the level so that the level is in its balance position at the beginning. As soon as we activate the motor, the lift force is generated and measured by the load cell. A mean lift force of 4.5g could be seen in the next figure.

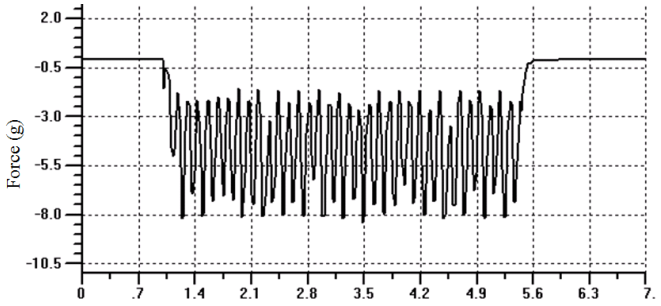


Figure 16: Force measurement

From the simple equation of equivalent moments, the lift force must be 0.4 time as much as the force measured by the load cell and therefore we get the mean value of 1.8 g. The mean lift measured from the experiment is a little bit bigger than one from the simulation, however it is still good to predict the total lift. Remind that the total mass of full prototype is 2.8 g, so it should fly if we employed two motors.

4.4 Take-off demonstration

A complete prototype was assembled with recommended configuration. It was let to move up and down unrestrictedly along two fixed parallel carbon tubes which, by the way, limit all other MAV's degree of freedom. At an input voltage of 4V peak-to-peak, the system starts to fly up. After a few seconds, a snapshot of the MAV motion (Figure 17) was taken, showing that it was above the initial position nearly 2 cm. It is not correlated to the take-off simulation due to the friction of the MAV with two carbon tubes.

5 CONCLUSION

In this study, a design tool based on a dynamic numerical model has been proposed to analyze the wing kinematic of a new MAV and thus predict the total lift force. An elastic element placed in parallel to the gearbox output shaft and wing allows operation at resonance. The effect of varying wing offset, elastic element stiffness, and elastic wing stiffness

have been simulated in order to determine the best configuration.

After experiments, we can conclude that an adequate amount of lift force is produced to bring our prototype to the air, which validates the results of our design tool. As the lift weight ratio is equal 1.28, it is not possible to handle electrical circuits including microcontroller, motor driver, Bluetooth device and also battery at this time. Future work has to be developed on increasing the working frequency but maintaining the same wing kinematic. If we succeed to double the working frequency the lift force will increase by the factor of four and the MAV has more spare weights.

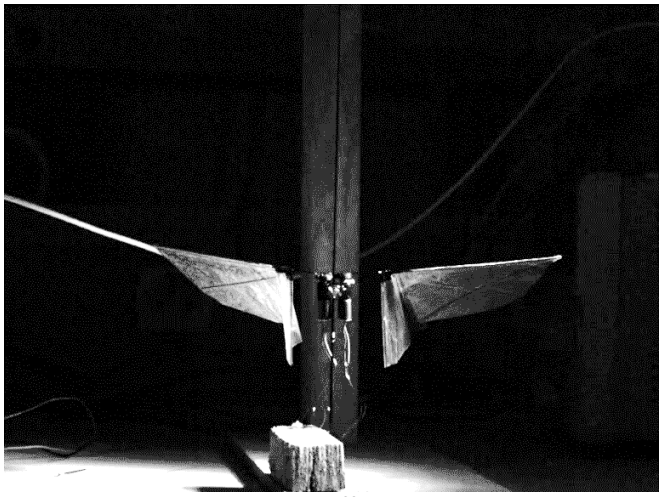


Figure 17: Take-off demonstration

APPENDIX

As Bond Graph model of the wing shown in this work is adapted from dynamic equations in [10], readers should refer to this reference for the full derivations. The following equation introduces only the complex Bond Graph elements related to the wing.

$$I_{flap} = \sin^2(\varphi)J_{22} + \cos^2(\varphi)J_{33} + \left((R_{CG} + d_w)^2 + \frac{\beta_{CG}^2}{2}\right)m_w - \frac{1}{2}\beta_{CG}^2\cos(2\varphi)\,m_w + \eta^2 J_m \quad (7)$$

$$I_{rot} = m_w\beta_{CG}^2 + J_{11} \quad (8)$$

$$M_{rot} = \ddot{\theta}(m_w(R_{CG} + d_w)\beta_{CG}\cos(\varphi) + J_{13}\cos(\varphi)) - \dot{\theta}^2\cos(\varphi)\sin(\varphi)\,(m_w\beta_{CG}^2 + J_{22} - J_{33}) - \vec{M}_{aero}.\vec{E}_1'' \quad (9)$$

$$M_{flap} = \ddot{\varphi}\cos(\varphi)\,(m_w(R_{CG} + d_w)\beta_{CG} + J_{13}) - \dot{\varphi}^2\sin(\varphi)\,(m_w(R_{CG} + d_w)\beta_{CG} + J_{13}) + \dot{\varphi}\dot{\theta}\sin(2\varphi)\,(m_w\beta_{CG}^2 + J_{22} - J_{33}) - \vec{M}_{aero}.\vec{E}_3 \quad (10)$$

Parameter	Value	Unit
K_s	2.956e3	mN.mm/rad
A	3.7	V
f	10	Hz
K_w	220	mN.mm.rad
d_w	35	mm
b_w	1.5	mN.mm.s/rad
$\varphi_{amplitude}$	$\pi\backslash 4$	rad
$\theta_{amplitude}$	1.5	rad
ϕ_{lag}	$\pi\backslash 2$	rad
F_{peak}	0.03	N

Table 1: Optimized parameter

ACKNOWLEDGMENT

This work was realized within the ANR CLEAR-Flight project 13-0012-001 (Controlled lift for efficient flight of an Artificial Insect) which is supported by the French program ANR ASTRID.

REFERENCES

[1] J. Song, H. Luo, and T. L. Hedrick, "Three-dimensional flow and lift characteristics of a hovering ruby-throated hummingbird," *Journal of The Royal Society Interface*, vol. 11, no. 98, pp. 20140541–20140541, Jul. 2014.

[2] S. P. Sane, "The aerodynamics of insect flight," *Journal of Experimental Biology*, vol. 206, no. 23, pp. 4191–4208, Dec. 2003.

[3] F.-O. Lehmann, "The mechanisms of lift enhancement in insect flight," *Naturwissenschaften*, vol. 91, no. 3, pp. 101–122, Mar. 2004.

[4] G. C. H. E. de Croon, M. Perçin, B. D. W. Remes, R. Ruijsink, and C. De Wagter, The DelFly. *Dordrecht: Springer Netherlands*, 2016.

[5] L. Ristroph and S. Childress, "Stable hovering of a jellyfish-like flying machine," *Journal of The Royal Society Interface*, vol. 11, no. 92, pp. 20130992–20130992, Jan. 2014.

[6] A. Bontemps, F. Valenciennes, S. Grondel, S. Dupont, T. Vanneste, and E. Cattan, "Modeling and evaluation of power transmission of flapping wing nano air vehicle," in *2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 2014, pp. 1–6.

[7] G. C. de Croon, M. A. Groen, C. De Wagter, B. Remes, R. Ruijsink, and B. W. van Oudheusden, "Design, aerodynamics and autonomy of the DelFly," Bioinspir. Biomim., vol. 7, no. 2, p. 025003, 2012.

[8] M. Keennon, K. Klingebiel, and H. Won, "Development of the Nano Hummingbird: A Tailless Flapping Wing Micro Air Vehicle," 2012.

[9] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood, "Controlled Flight of a Biologically Inspired, Insect-Scale Robot," Science, vol. 340, no. 6132, pp. 603–607, May 2013.

[10] L. Hines, D. Campolo, and M. Sitti, "Liftoff of a motor-driven, flapping-wing microaerial vehicle capable of resonance," *Robotics, IEEE Transactions* on, vol. 30, no. 1, pp. 220–232, 2014.

[11] W. B. Dickson, A. D. Straw, C. Poelma, and M. H. Dickinson, "An integrative model of insect flight control," in *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006, pp. 31–38.

Quad-thopter: Tailless Flapping Wing Robot with 4 Pairs of Wings

Christophe De Wagter*and Matěj Karásek†and Guido de Croon‡
Micro Air Vehicle Laboratory, Delft University of Technology, Kluyverweg 1, The Netherlands

ABSTRACT

We present a novel design of a tailless flapping wing Micro Air Vehicle (MAV), which uses four independently driven pairs of flapping wings in order to fly and perform agile maneuvers. The wing pairs are arranged such that differential thrust generates the desired roll and pitch moments, similar to a quadrotor. Moreover, two pairs of wings are tilted clockwise and two pairs of wings anti-clockwise. This allows the MAV to generate a yaw moment. We have constructed the design and performed multiple flight tests with it, both indoors and outdoors. These tests have shown the vehicle to be capable of agile maneuvers, and able to cope with wind gusts. The main advantage is that the proposed design is relatively simple to produce, and yet has the capabilities expected of tailless flapping wing MAVs.

1 INTRODUCTION

Flying animals remain unrivaled when it comes to their flying skills and flight characteristics. Hummingbirds can hover and maneuver in narrow spaces to feed and then subsequently fly hundreds of kilometers when migrating [1]. Besides the energy and sensory processing aspects, a great deal of the advantages of flying animals over current Micro Air Vehicles (MAVs) are attributed to their way of propulsion. Flapping wings are predicted to achieve higher lift coefficients than conventional MAV designs, especially when scaled further down towards insect scales. In addition, they are expected to have a higher energy efficiency when flying at higher speeds, extending range and duration of the flight [2].

Despite considerable efforts - and successes [3, 4] - in the last few decades, the dominating MAV types are still rotorcraft, fixed wings or recently combinations of both[5, 6]. A main reason for this is the difficulty of producing a flapping wing MAV that fulfills some of the promises of animal flight.

On the one hand, there is a large class of 'tailed' flapping wing MAVs, which goes back to rubber-band flapping wing vehicles designed in the 19th century [7]. Flapping wing MAVs such as 'small bird' [8], 'big bird' [9], or the 'DelFly' [10], have single degree of freedom motor-driven flapping wings for generating thrust. The control moments are generated by actuated control surfaces on the tail. Since the tail is relatively large, it dampens the body dynamics sufficiently to make this type of MAV passively stable.

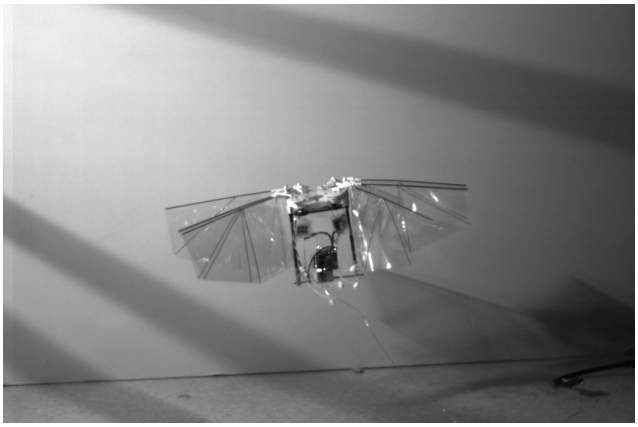


Figure 1: High speed camera recording of a quad-thopter

The tail actuation typically consists of a rudder and an elevator, and can be used for changing the MAV's direction, height, or velocity. However, the aerodynamically stabilizing tail section also makes the vehicle particularly sensitive to external perturbations [10]. The forces and moments generated by the tail actuators are in general insufficient to compensate perturbations in 'gusty' environments, with even air-conditioning causing considerable problems to these light wing loading MAVs. Finally, elevator and rudder effectiveness vary dramatically based on the incoming airflow and can even reverse when descending in hover. This makes tuning autopilot control loops dependent on more sensors and creates uncontrollable areas in the flight envelope.

On the other hand, there is a growing class of 'tail-less' flapping wing MAVs, which use the wings themselves for control. The idea is that the wings can generate much larger forces and moments in shorter times than tailed actuators. In combination with the absence of tail and its damping effect, this leads to a higher maneuverability. The first successful design of this class was the 'Nano Hummingbird' [3]. It featured an ingenious but complex mechanism to generate all three moments required for full attitude control. Recently, other MAVs of similar size have been designed, which aim for simpler designs, but which have not yet shown the same

*Email address: c.dewagter@tudelft.nl
†Email address: m.karasek@tudelft.nl
‡Email address: G.C.H.E.deCroon@tudelft.nl

maneuverability as the Nano Hummingbird and, at the same time, suffer from very limited flight endurance of several tens of seconds at best [11, 12, 13]. The smallest type of flapping wing MAV of this class is the well-known 'Robobee' [14], which for now requires the energy source to be off-board.

Although current tail-less flapping wing MAVs are closing in on the ideal set by nature, none of them are yet both able to perform real flight missions and at the same time relatively easy to construct.

To broaden the field of application of flapping wing MAVs, a light and simple wing actuation mechanism would be needed that can quickly create large attitude control moments in all three axes. Based on this idea, we present in this paper a new tailless flapping wing MAV design, referred to as a 'quad-thopter'. The design is similar to a quadrotor, in the sense that it uses the thrust of four wing pairs to do thrust vectoring (Figure 1). It is also reminiscent of the very early 'Mentor' design [15], which also had four wing pairs for flying. However, that design used a single main actuator driving the 4 wings at the same flapping frequency. The control relied upon control surfaces interacting with the wake of the flapping wings, which had rather low effectiveness, limiting the controllability of the system. Instead, the 'quad-thopter' can drive all wings independently from zero to maximal thrust, which can generate significant roll and pitch moments, and the flapping planes of diagonally opposing wing-pairs are tilted with respect to each other for yaw controllability.

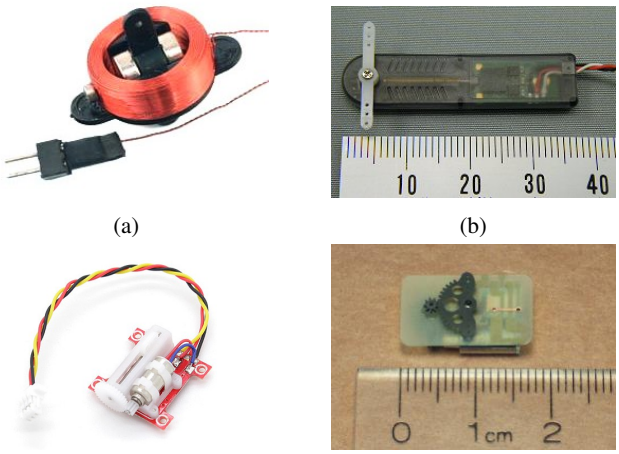
The quad-thopter design proposed in this paper represents a close-to-optimal choice in the design space consisting of the magnitude of the generated control moments, the control bandwidth, and the weight, size and energy requirements of the actuators. In addition, the quad-thopter is relatively easy to construct with widely available current-day technology, and has a flight time of 9 minutes or more, depending on the flight regime. Hence, it is suitable for real-world missions.

In Section 2, we discuss current flapping wing designs and actuators in more detail, in order to get a better understanding of the difficulties involved in tailless flapping wing MAV design. Then, in Section 3, we present the new design. We study the body's vibrations in Section 4 and the less evident yaw moment generation in Section 5. We describe the flight characteristics in Section 6, showing pictures of the flapping wing MAV in flight and providing links to flight footage. Finally, we draw conclusions in Section 7.

2 Tail-less flapping wing

2.1 Moment generation

Most ornithopter designs use a tail, which provides passive aerodynamic stabilization and typically carries also conventional actuated control surfaces. When the tail is removed, active stabilization becomes necessary and some mechanism is required to create the 3 moments needed to orient and stabilize the platform.



(a) (b) (c) (d)

Figure 2: Overview of actuator types for lightweight flapping wing MAVs: (a) magnetic servos, (b) shape memory alloy servos and (c)(d) servos with brushed DCs (images from www.microflight.com,www.servoshop.co.uk, www.hobbyking.com, www.microflierradio.com).

Many solutions have been proposed. Some add propeller thrusters besides the flapping wing [16]. But the vast majority of researchers, inspired by biological fliers, search for new degrees of freedom to incorporate in the main flapping wings to vary their aerodynamic force over the flapping cycle [3, 4, 17, 13]. To use these degrees of freedom in closed loop control, they must be actuated with sufficient speed and force.

2.2 Hovering without tail

The minimal requirement for controllable hovering of an aircraft is thrust vectoring. Instead of controlling the 6 DOF (3D position and 3 attitude angles) of the free flying body directly, 2 position variables are controlled indirectly through the attitude which in turn controls the thrust vector and hereby the longitudinal and lateral acceleration. This allows for 6DOF hover with only 4 independent control variables. Most concepts use flapping power control combined with 3 external actuators – for instance to move the roots of trailing edges [18] or drive all the flapping degrees of freedom [17]. Since actuators do not contribute to thrust generation but only add weight, these must be very light. Finding sufficiently light, fast and strong actuators is an integral part of designing a flight-capable multi degree of freedom flapping mechanism.

2.3 Actuator Review

The main driving motor must be sized to produce sufficient thrust, but sizing the control actuators is more complex. In practice, on small flapping wing vehicles in the presence of disturbance, actuators must be fast, strong and light. This combined requirement is not trivial.

Coil actuators (Figure 2 (a)) are fast, but create very small

moments, which makes them suitable only for actuation of conventional tail control surfaces. Shape memory alloys (Figure 2 (b)) have shown high strength at minimal weight, but are slow, fragile and create minimal deflections, that need to be amplified.

Most servos consists of small brushed motors with a reduction gearbox and include a position feedback mechanism with a potentiometer (Figure 2 (c)) or magnet and hall effect sensor (Figure 2 (d)). The gear ratio can be altered to change the speed versus force, but to increase both, a larger and heavier motor is needed; its size can even come close to the one of the main flapping motor. In contrast with the main motor which runs all the time, actuator motors are used very inefficiently and only work part of the time.

2.4 *Moment control using the flapping motor*

To use most of the actuators in their efficient regime, main flapping actuator(s) can be used to also generate the control moments. Such idea is not novel. RoboBee [14] uses the two main flapping piezo-actuators driven with independent waveforms to generate the 4 independent controls (See Figure 3 (a)). The flapping amplitudes of the left and right wings can be driven independently, and a bias can be added (to both actuators) for pitch control. Finally a speed difference in up- and down-stroke can generate yaw moments, while the same flapping motion also provides the main thrust force.

The quest to achieve this same idea using traditional rotating electric motors has led some researchers to attach brushed motors directly to the wings [19] as illustrated in Figure 3 (b). These motors are used outside their design operational regime with very low efficiency and high wear as they vibrate back and forth instead of turning in one direction at high speed. Nevertheless, their efficiency can be improved by using resonance mechanisms. All 3 required control moments can be generated by varying amplitude of the stroke and velocity profiles within the stroke in a differential way (left/right and upstroke/downstroke).

Still, electric motors are most efficient when turning at higher speed, in which case a crank mechanism is required. Unless a variable crank mechanism is used—which in turn is controlled by actuators—this makes it impossible to vary amplitude anymore while also the phase and frequency become coupled.

To generate different thrust on the left and right wings, they must be uncoupled and driven by separate motors. In this case, the motors are used efficiently, since their main task remains to be thrust generation, while variations anywhere between zero and full power can yield very large moments with minimal response times. This, however, comes at a cost that it is impossible to keep both wings in phase.

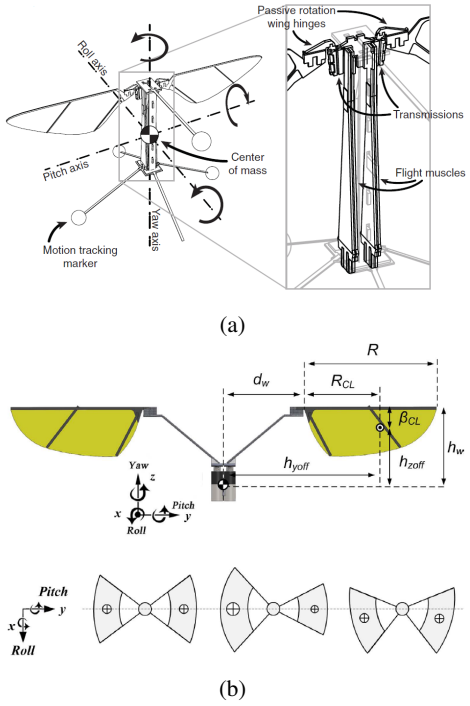


Figure 3: MAV designs that use their main actuators also for control: (a) piezo actuators [14] and (b) brushed DCs [19].

3 The Quad-thopter

In order to have full control authority in hover, which requires independent generation of at the three body moments and the total thrust, one solution is to combine four sets of wings, each driven by a separate motor and a crankshaft as is shown in Figure 4. When the four thrust vectors can be controlled independently, this can generate moments for attitude control much like a quadrotor, allowing full 3D hover control.

But unlike in a quadrotor, where propellers have a non-zero average torque, an additional control is needed for the

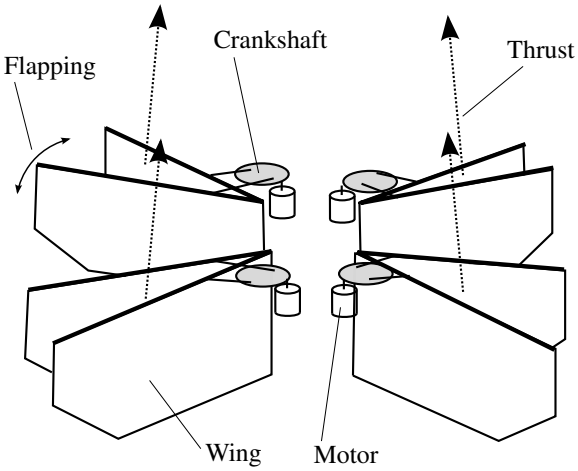


Figure 4: Quad-thopter. Four pairs of flapping wings are arranged in an X-configuration with a small angle between thrust vectors to allow control of the yaw axis.

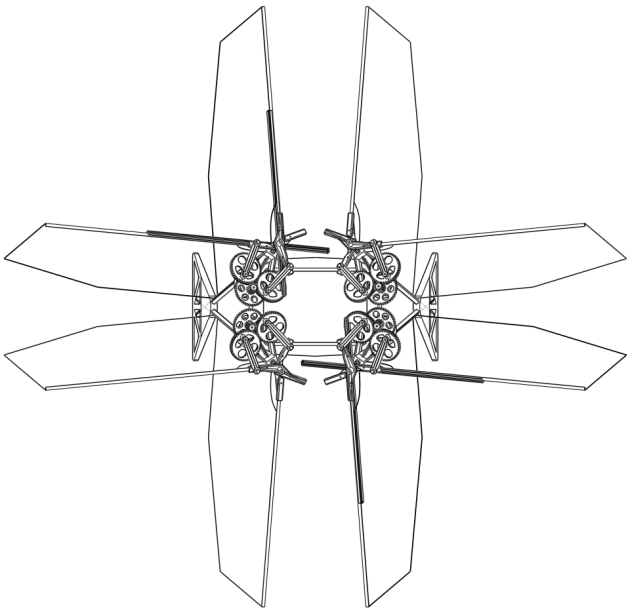


Figure 5: Quad-thopter Top View. When thrust vectors are non-parallel, two oping pairs of wings can create a yaw moment. The maximal dimension is 28 cm from tip to tip and the weight is 33 gram.

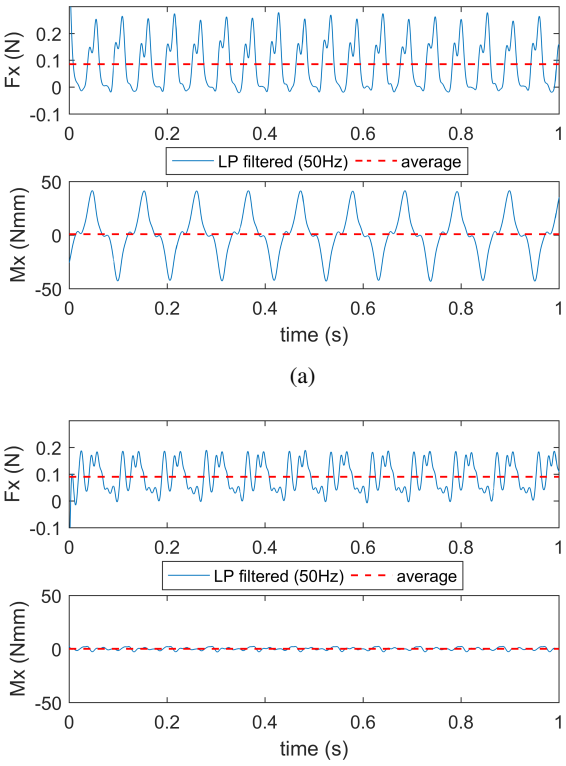


Figure 6: Thrust force and moment around principal body axis (data include also inertial effects): (a) single wing flapping with 90 degree amplitude, (b) double-wing flapping in anti-phase with 40 degree amplitude. The reaction torque on the body is significantly reduced when using the double-wing setup while generating similar amount of thrust as the single wing.

yaw. This can be obtained by tilting the thrust vectors with respect to the average thrust vector as per Figure 5.

This setup does still suffer from the effect described in Section 2 that wings can flap out of phase. This could potentially lead to very large yawing moments on the fuselage, resulting in fuselage rotation that will cause loss of flapping amplitude and loss of lift. To cope with this problem, instead of using single flapping wings, a phase locked pair of wings as found in for instance the DelFly II [10] is used instead. This means that whatever frequency each of the four motors is running, for each single wing moving one way there is a corresponding wing moving the other way, canceling each other out.

The resulting setup has fast and powerful attitude control while its complexity remains moderate. On the one hand, four gearboxes are needed, but on the other hand, simple fixed gear crankshafts can be used. Fragile, underpowered, slow or expensive actuators are no longer needed. In terms of weight, all actuators are directly used to create thrust, which increases efficiency and the maximal available thrust.

The lack of tail section significantly reduces the sensitivity for perturbations, while active attitude control with full authority controls the attitude. This enables maneuvers that were not possible with the tail, like a fast vertical descend.

The platform is capable to transition to forward flight in the same way as its tailed counterpart. In forward flight attitude must also be actively controlled. Similarly as with hybrids like the Quadshot [5], the vehicle pitches down almost 90 degrees and the wings start to produce lift perpendicularly to the thrust direction.

4 Residual Vibration

Although the moments of the flapping itself are canceled out during stationary hover as shown in Figure 6, the thrust generated by a wing pair is non constant in time. The fact that all wings generate thrust and flapping-torque with peaks at different times still results in vibrations on the main central fuselage.

The DelFly concept has been using a double pair of flapping wings to minimize fuselage rocking. For every wing performing an upstroke there is exactly one wing doing a downstroke. The double pair of wings doing clap and fling has also shown to be able to achieve higher thrust density [10].

This concept can be re-used in the tail-less flapper with 4 wings and 4 motors. Replacing every wing with a pair of in anti-phase flapping wings removed the largest residual vibra-

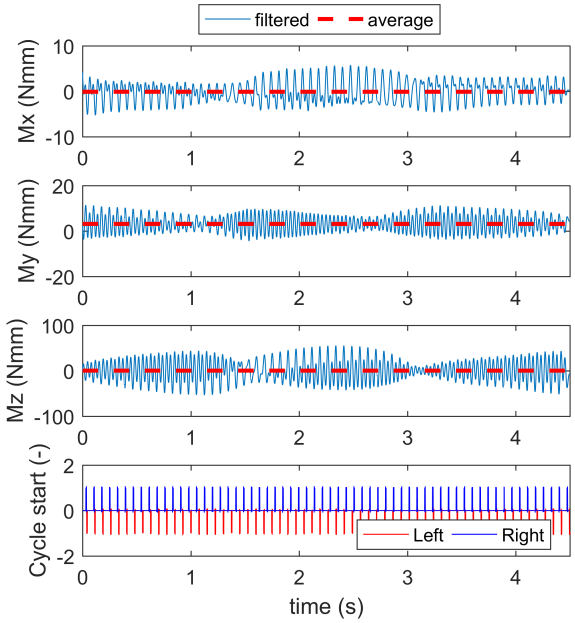


Figure 7: Two double-wing experiment: beat phenomenon can be observed in the moment data when a difference in flapping frequencies of left and right double-wings is present. The cycle start is detected by a hall effect sensor and a magnet attached to the flapping mechanism. The residual vibration is especially strong around the roll (z) axis.

tion. The wing mass in this case does not cause large inertial vibrations anymore, because for any wing moving in one direction another wing moves in the opposing direction.

The result is a vehicle with 4 main driving motors and 4 pairs of flapping wings flapping at different rates. The main residual vibration now is when 2 opposing pairs flap with 90 degrees phase shift, with the difference between the minimal thrust during a stroke and the maximum thrust during a stroke as the driving force for the vibration. Due to their different rates, the phase shift is not constant, but varies over time; a beat phenomenon (vibration of pulsating amplitude) will be present, see Figure 7. When using a wing design with small thrust variation during a stroke, this vibration can be reduced to acceptably small levels.

To keep fuselage motion to a minimum, fuselage inertia $I = m \cdot r^2$ can play an important role.

5 Yaw versus thrust efficiency

Pitch and roll are driven by differences in thrust generated by the left and right wings and fore and aft wings, respectively, but yaw is less evident. To achieve yaw, the lift vectors of 2 opposing wings are misaligned with respect to vertical body axis. One diagonal is given a right-hand yawing alignment while the other pair of wings is given a left-hand yawing moment.

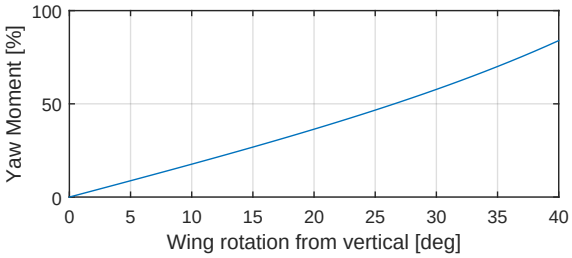


Figure 8: Yaw force in function of thrust rotation. Note that the yawing moment increases more than linearly with the wing rotation due to the average hover-lift increase caused by the efficiency loss.

The amount of misalignment can be used to increase the yaw control effectiveness at the cost of less efficient thrust generation as not all lift vectors now point perfectly upward.

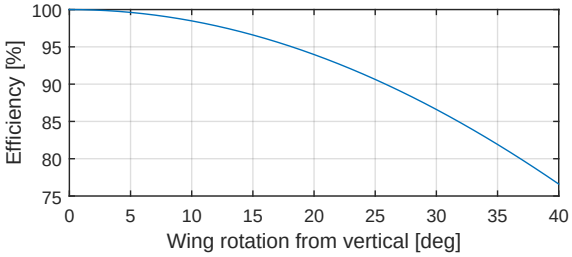


Figure 9: Efficiency in function of thrust rotation.

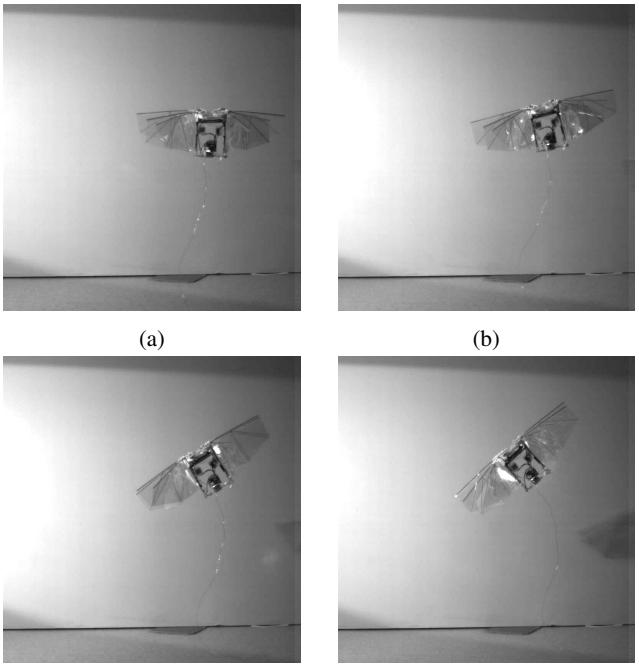
Since thrust efficiency is lost to achieve yaw control, the yaw channel could still benefit from using an actuator instead. Since the yaw is very well damped thanks to the wing area, a slower but more powerful actuator could still be considered to for instance deflect the trailing edges of the wing [18] to also deflect the thrust vector. In this case, only three sets of flapping wings would be required for full attitude control much like the tricopter concept.

6 Flight Testing

A quad-thopter was built using DelFly II flapping mechanisms. Instead of a double pair of wings, only one side was mounted. DelFly II brushless motors were used and equipped with 3.5 Amp BLDC motor controllers. Since the vehicle is not naturally stable a paparazzi-UAV [20] Lisa-S [21] autopilot was mounted. Standard rotorcraft stabilization was programmed and the Quad-thopter was tuned during manual flight in attitude direct mode.

Figure 10 shows the response to a 40 degree step input in roll. Within less than 4 beats of the fastest flapping wings (15 Hz) the attitude change was fully obtained.

Position step responses were performed and measured using an Optitrack camera system. The quad-thopter was commanded in attitude mode to make a lateral step of about 2



(a) (b) (c) (d)

Figure 10: Highspeed camera recordings at 66.6 ms interval show a step in attitude from hover to a steady 40 degrees of roll being executed in less than 266 milliseconds or less than 4 wing beats at 15 Hz.

meters. A side view of the maneuver is shown in Figure 11. The quad-thopter will quickly reach the commanded left roll angle of 50 degrees and start accelerating. About half a meter before the target, the attitude is commanded to zero. Because of the lateral area of the wings and relatively low wing loading, the quad-thopter stops by itself when commanded back to zero attitude. Then a right step is commanded. Everything combined is executed in under 3 seconds.

The corresponding timing of the motion is shown in Figure 12. As can be seen, the entire lateral acceleration from hover followed by 2 meter motion and deceleration only takes about one second. Figure 13 shows the roll angle of the quad-thopter during the maneuver. It shows that roll angles of over 50 degrees are achieved in about a quarter of a second. Finally the speed profile of the lateral step is shown in Figure 14. Please note that during the lateral step the quad-thopter was only rolled 50 degrees and did not nearly reach its maximum speed but instead was subjected to lateral drag.

Lateral steps at higher angles were performed but often resulted in lost tracking from the Optitrack. One sequence at 80 degrees roll was successfully recorded during a 3m lateral step as shown in Figure 15. As shown in Figure 16 the quad-thopter reaches speeds of 3.5 m/s and roll angles of 80 degrees while stepping sideways 3m in less than 1.5 seconds.

To illustrate the forward flight and disturbance handling

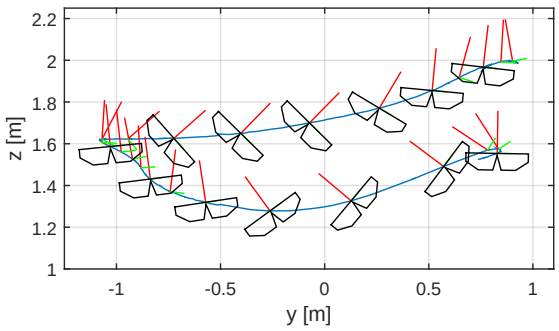


Figure 11: Indoor test flight recorded by Optitrack. The quad-thopter starts at the bottom right and makes a 2m step to the left and then back to the right in under 3 seconds. Notice that the vehicle does not need negative roll during the slow down.

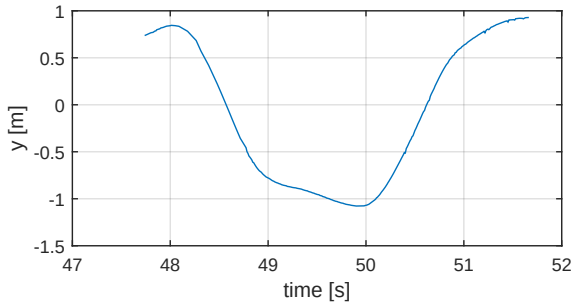


Figure 12: Lateral position change in function of time during the lateral step shown in Figure 11.

capabilities, outdoor flights have been performed as shown in Figure 17. Very aggressive start and stops are possible. when compared to DelFly II with an aerodynamic tail, the sensitivity to turbulence is reduced an order of magnitude by the fast powerful moments from the electronics attitude control using wing pairs. The maximal flight speed however is very close to that of DelFly II and is limited by the maximal flapping frequency that can be obtained.

Video footage of quad-thopter flight were placed on YouTube[1].

7 Conclusions

In this paper, we proposed a novel flapping wing design, a 'quad-thopter'. In the article, we have discussed the various design parameters relevant to a highly maneuverable, tail-less flapping wing MAV. We conclude that the design represents a close-to-optimal choice in the design space consisting of the magnitude of the generated control moments, the con-

[1] https://www.youtube.com/playlist?list=PL_KSX9GOn2P9HTG4SY59KbgH2fT9cxY06

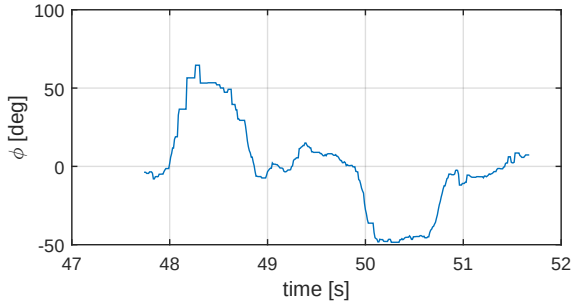


Figure 13: Roll angle during lateral step.

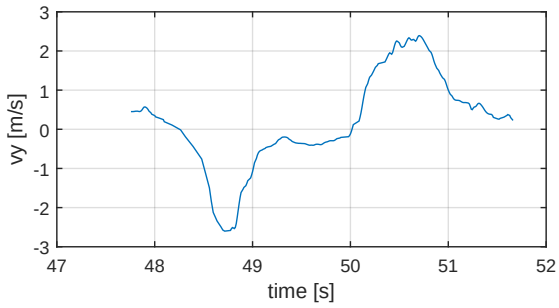


Figure 14: Speed during lateral step.

trol bandwidth, and the weight, size and energy requirements of the actuators. In addition, the quad-thopter is relatively easy to construct with widely available current-day technology. The implementation of the design built and tested in this work has a flight time of 9 minutes or more, depending on the flight regime. This makes it suitable for real-world missions.

Although the presented design does not correspond to any (known) biological counterpart, the quad-thopter has a number of characteristics featured by natural fliers. For instance, the proposed quad-thopter becomes more efficient in forward flight, much more than quadrotors, increasing the range and endurance. Furthermore, the wing surfaces induce

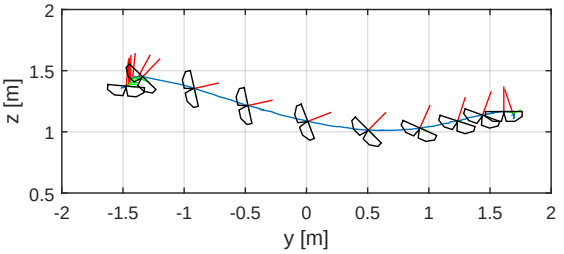


Figure 15: A 3m lateral command where speeds of 3.5m/s and angles of 80 deg roll are reached.

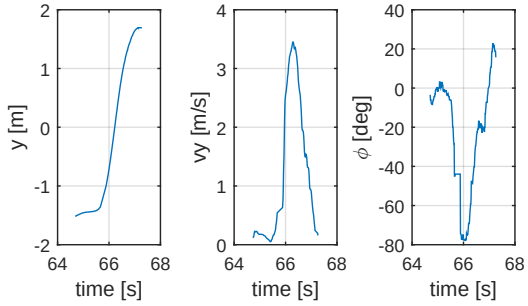
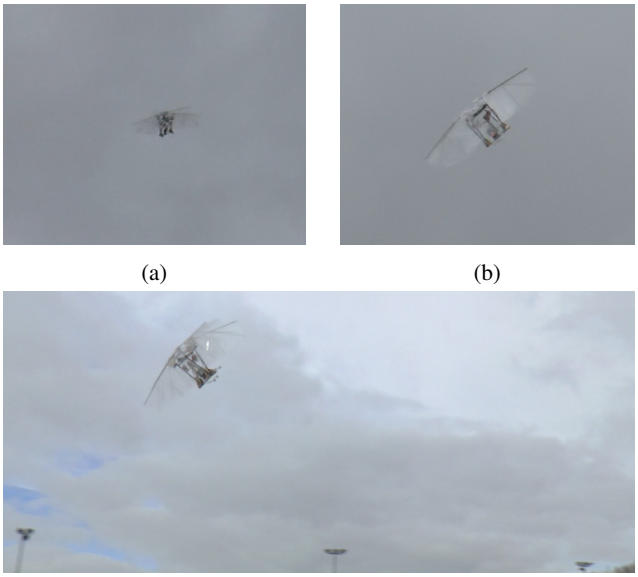


Figure 16: A 3m lateral command.



(a) (b)

(c)

Figure 17: Quad-thopter In-flight Outdoor in various phases of the flight. (a) hover (b) semi-transitioned (c) fast forward flight.

drag, which can be used for braking. This means that in contrast to quadrotors, quad-thopters do not have to thrust in the backward direction to brake, which also gives them the ability to brake faster. Finally, the quad-thopter features an enhanced safety because of the absence of fast-rotating rotors, so it is more suitable for flight around humans.

We hope that the presented design will be more apt than previous designs for wide-spread use in academia and industry, helping to break the hegemony of rotorcraft and fixed wings.

References

[1] Anders Hedenström. Extreme endurance migration: what is the limit to non-stop flight? *PLoS Biol*, 8(5):e1000362, 2010.

[2] Max F Platzer, Kevin D Jones, John Young, and JC S. Lai. Flapping wing aerodynamics: progress and challenges. *AIAA journal*, 46(9):2136–2149, 2008.

[3] M. Keennon, K. Klingebiel, H. Wonc, and A. Andriukov. Tailless flapping wing propulsion and control development for the nano hummingbird micro air vehicle. In *American Helicopter Society Future Vertical Lift Aircraft Design Conference, San Francisco, California*, 2012.

[4] Robert Wood, Radhika Nagpal, and Gu-Yeon Wei. Flight of the robobees. *Scientific American*, 308(3):60–65, 2013.

[5] Pranay Sinha, Piotr Esden-Tempski, Christopher Forrette, Jeffrey Gibboney, and Gregory Horn. Versatile, modular, extensible vtol aerial platform with autonomous flight mode transitions. In *IEEE Aerospace Conference*, March 2012.

[6] C. De Wagter, R. Ruijsink, E.J.J. Smeur, K. van Hecke, F. van Tienen, E. v.d. Horst, and B. Remes. Design, control and visual navigation of the delftacopter. *arXiv preprint arXiv:1701.00860*, 2017.

[7] O. Chanute. *Progress in Flying Machines*. Dover, 1894, reprinted 1998.

[8] W. Bejgerowski, A. Ananthanarayanan, D. Mueller, and S.K. Gupta. Integrated product and process design for a flapping wing drive-mechanism. *ASME Journal of Mechanical Design*, 131, 2009.

[9] J.W. Gerdes, S.K. Gupta, and S. Wilkerson. A review of bird-inspired flapping wing miniature air vehicle designs. *Journal of Mechanism and Robotics*, 4(2), 2012.

[10] G. C. H. E. de Croon, Mustafa Percin, B. D. W. Remes, Rick Ruijsink, and C. De Wagter. *The DelFly - Design, Aerodynamics, and Artificial Intelligence of a Flapping Wing Robot*. Springer Netherlands, 2016.

[11] Inderjit Coleman, David and Benedict, Moble and Hrishikeshavan, Vikram and Chopra. Design, Development and Flight-Testing of a Robotic Hummingbird. In *AHS 71st Annual Forum, Virginia Beach, Virginia, May 57, 2015*, page 18, 2015.

[12] Hoang Vu Phan, Taesam Kang, and Hoon Cheol Park. Design and stable flight of a 21g insect-like tailless flapping wing micro air vehicle with angular rates feedback control. *Bioinspiration & Biomimetics*, 12(3):036006, apr 2017.

[13] A Roshanbin, H Altartouri, M Karásek, and A Preumont. Colibri: A hovering flapping twin-wing robot. *International Journal of Micro Air Vehicles*, 0(0):1756829317695563, 0.

[14] K.Y. Ma, P. Chirarattananon, S.B. Fuller, and R.J. Wood. Controlled flight of a biologically inspired, insect-scale robot. *Science*, 340(6132):603–607, 2013.

[15] P. Zdunich, D. Bilyk, M. MacMaster, D. Loewen, J. DeLaurier, R. Kornbluh, T. Low, S. Stanford, and D. Holeman. Development and testing of the mentor flapping-wing micro air vehicle. *Journal of Aircraft*, 44(5):1701–1711, 2007.

[16] WowWee. Flytech dragonfly. Wikipedia entry, accessed May 16, 2017.

[17] Matej Karasek, Alexandre Hua, Yanghai Nan, Mohamed Esseghir Lalami, and André Preumont. Pitch and roll control mechanism for a hovering flapping wing mav. In *International Micro Air Vehicle Competition and Conference 2014*, pages 118–125, Delft, The Netherlands, August 2014.

[18] Woei Leong Chan, Quoc Viet Nguyen, and Marco Debiasi. Tailless control of a double clap- and-fling flapping wing mav. In Prof. Zhihong PENG and Dr. Feng LIN, editors, *International Micro Air Vechicle Competition and Conference 2016*, pages 291–299, Beijing, PR of China, okt 2016.

[19] Lindsey Hines, David Colmenares, and Metin Sitti. Platform design and tethered flight of a motor-driven flapping-wing system. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5838–5845. IEEE, may 2015.

[20] Pascal Brisset, Antoine Drouin, Michel Gorraz, Pierre-Selim Huard, and Jeremy Tyler. The paparazzi solution. In *MAV 2006, 2nd US-European Competition and Workshop on Micro Air Vehicles*, Sandestin, United States, October 2006.

[21] BDW Remes, P Esden-Tempski, F Van Tienen, E Smeur, C De Wagter, and GCHE De Croon. Lisa-s 2.8 g autopilot for gps-based flight of mavs. In *IMAV 2014: International Micro Air Vehicle Conference and Competition 2014, Delft, The Netherlands, August 12-15, 2014*. Delft University of Technology, 2014.

Analysis of Folding Wing Rolling Moment

N. Krajangsawasdi, T. Pantuphag,S. Catteeyothai, and C. Thipyopas*
Department of Aerospace Engineering, Faculty of Engineering, Kasetsart University, Bangkok, 10900, Thailand

ABSTRACT

Mini and micro aircraft are required to be a multi functioning; which can both maneuver in forward flight and hover for takeoff configuration. This leads to the development of the folding wing aircraft. The folding wing cannot apply any control surface to create a rolling movement because it must move all the time. The benefit of folding is an idea for rolling moment generation. This is an asymmetrical lift that generates form folding unequal area between left and right wing. Therefore, this research focuses on finding a relationship between rolling moment generated by the asymmetrical force and the area ratio of two wings by using CFD simulation to predict them. Firstly, the design of the wing is simulated in cruise condition at various angles of attack to determine the trim angle and the significant aircraft characteristics. Then the trim angle was set at the condition for computing the rolling moment for 3 levels of wing folding. The three levels are not too different in overall wing area, but the area of the two sides are imbalanced so that can generate different rolling moment. The result of the simulation shows that the asymmetry of the wings can generate a rolling moment and it increases dramatically when the area ratio rises.

1 INTRODUCTION

Generally, there are three control axes on the aircraft that are longitudinal, lateral and vertical axis that are called roll, pitch and yaw respectively. Nowadays, the mini and micro aircraft developers desire the multi-mission aircraft that can do a various task [1, 2]. That is the reason to develop the wing that can fold. It is easy to transport and that can also achieve a high performance vertical takeoff because of less drag from the big wing area when hovering. According to the folding condition, the conventional rolling control surface; aileron, on the normal fixed wing is difficult to install. This is a challenge to research and develop a new process to control rolling motion. The object of this research is to determine the relation between different of the wing area when folded and the rolling moment that is generated by using Computational Fluid Dynamics (CFD) to find all of the aircraft parameters and compare them to the original fixed wing which was named Brown that are designed for multi-missions that can maneuver in forward flight and hovering in takeoff configuration. The developed folding wing must replace the original in the same or better performance. There are many ways to roll the aircraft but the principle of rolling control is to generate asymmetric lift between two wings. For example, Bamber [1934] [3] created some new components called Floating Wing Tip Aileron which is installed at the end of the wings to make a rolling moment to escape a stall situation, Next Rao [1983] [4] attached an additional control surface to the leading edge. When it works, the position of lift force acting on each side is changed and counteracts any instability. In the progression of material science in the 21st century, Raney [2000] [5] makes an effort to use the smart material to make the flexible wing for adapt its shape to be the suitable shape of rolling motion. Furthermore, Ifju [2005] [6] claimed that there are a plenty of benefits from the wing which can flex same as to the natural way. The result of wind tunnel experiment and CFD are the same trend that shows the flexibility provides for smoother flight than conventional wings. They do not have only good maneuverability and also can delay stall. It can be seen that the method attempted was to follow a natural way like how birds can flex their wings. The simulation of the computation of fluid mechanics was to set the flow as laminar flow and increase the number of elements. This method tries to avoid using turbulence flow which makes the calculation more difficult.

2 METHOLOGY

This analysis was divided into 4 parts. First of all, the complex wing was simplified to be a thin plate that is 2 mm thick but the platform is similar to the original. Secondly, there is a determination of grid independence in the Flow Simulation Program by increasing the number of elements within the model. Then, the simplified wing was simulated in the program to calculate the aircraft characteristics of the wing. Eventually, the computational process of simplified folded wings was simulated to find the moment coefficient when the wing was folded. However, there are some information about aircraft and setting the simulation program.

2.1 Aircraft specification and other parameters

Some significant parameters and aircraft performance of the original aircraft (Brown) were assigned by the developer as shown in table 1. This new folding wing must follow the maximum take of weight (MTOW), lift coefficient and rolling coefficient. Wing platform was design from the shape of birds wing, so it is somewhat of strange platform. The wing diameter is shown in Fig.1.

*Email address: fengcpt@ku.ac.th

MTOW	Vcruise	S;m2	C;m	b;m	CL	Cl
0.5 kg	15 m/s	0.13	0.25	0.48	.274	.00349

Table 1: The aircraft parameters defined by the original aircraft.

2.2 *Analysis of the force and moment data*

The simulation model was set axis follow by the aircraft principal axis. They have 3 axes; longitudinal, lateral and vertical axis that are represented by letter X, Y and Z respectively. The direction of the axis is shown in Fig.1 and its origin is assumed at the center of gravity of the wing. Moreover, this point position and the axis in another model is same as this wing.

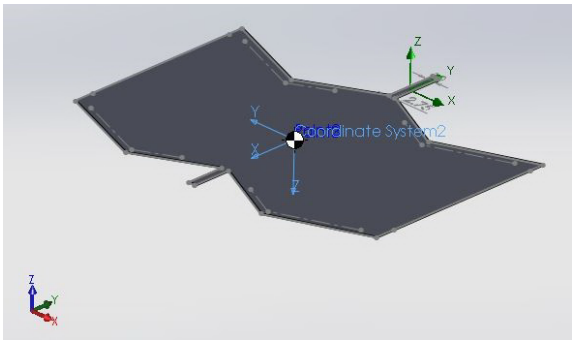


Figure 1: Setting axis for simulation.

If there is some angle of attack between wing and wind direction, lift is not the Z-direction force, which can read from simulation result, so it must be calculated by vector method. Lift and drag force are shown in Fig.2 which are the composition of the X and Z force.

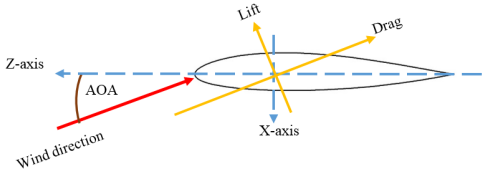


Figure 2: Aircraft axis and aerodynamic force axis.

2.3 *Simplified wings*

The actual folding wing has a lot of components such as bolts, nuts, wing axis and plates. If the simulation takes all of the wing components into account, it will be waste of resource and also too complicated to compute. Thus, the wing that have a plenty of plates was simplified to be normal wing that is a thin plate in the original wing planform. The simplified wing in cruise condition is shows Fig.3.

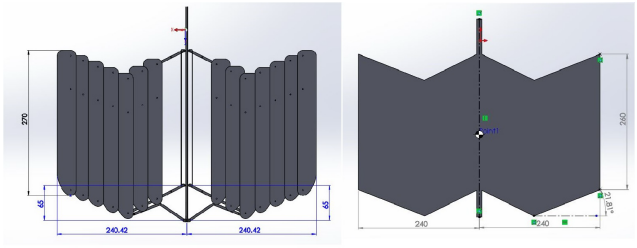


Figure 3: Comparison between actual (left) and simplified (right) plain wing.

2.4 *Domain dependence*

Domain is the region where the simulated pass. The domain size is one of the factor that tell how fast the analysis. If this is an enormous domain size, there is waste of time. Following this, the suitable domain size should be calculated. First of all, the full domain size 8 times of chord that 4 m wide along left to right, 2 m long in back, 1 m in front and 2 m high was set as the first trial domain size. Secondary, there is an idea to make the domain to be symmetry because it may reduce calculation time. This lead to symmetry domain that use the size 8 times same as the first one, but it computes only left wing. This usage time is haft of the first domain. Then, the bigger domain was set to check the flow behavior. This was defined as 20 times of chord. The result of this calculation is not noticeably difference from the previous calculation. The lift coefficient of each domain and show in table **??** the error in this table compare with the original required lift coefficient ,0.274. Overall, the 8-time-chord domain dimension was chosen to the next step simulation because the error of lift coefficient is under 5 percentage and the usage time is minimal.

Domain	Lift coef.	Drag coef.	error
(2+2)x(2+1)x(1+1)	0.2764	0.0385	0.90
(2+0)x(2+1)x(1+1)	0.2876	0.0394	4.99
(5+0)x(5+2.5)x(2.5+2.5)	0.2789	0.0387	1.80

Table 2: Result in lift coefficient for different domain sizes. Domain:(left+right)x(back+fornt)x(top+bottom)

2.5 *Set up simulation model for plain wing*

For the plain wing analysis, the wing is symmetrical, so it can be computed only half wing by using the symmetrical domain to reduce the number of element. The domain dimension (in section 2.4) is 3 m long (1 m front and 2 m back), 2 m wide with the symmetric half wing and 2 m height. Other flight conditions were assigned to the atmosphere and the wind velocity magnitude of 15 m/s and the direction was changed by the vector component method shown in Y and Z direction followed by the adjustment angle of attack.

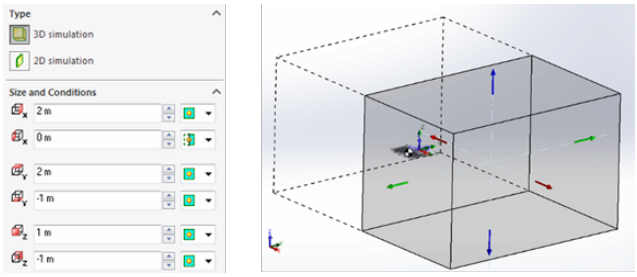


Figure 4: Computation domain.

2.6 *Grid dependence*

The original aircraft with fixed wing was designed for carry 0.5 kg total weight and the angle of attack at 6-degree cruise condition. Therefore, this condition is set to be the validation condition that grid is meshed fine enough to compute lift force equal to the carrying weight. Grid independence is the process to find the minimal element number that will give the best result. In addition, it can avoid to waste computer resource. Firstly, the model was meshed in the coarse level and simulate to measure the lift force to compare with the original plane condition, afterwards it was meshed in the finer level (coarse and fine mesh are shown in figure 5). Following that, lift coefficient for each mesh levels were calculated each mesh level and they were plot (figure 6) and find the different between the coarse and the finer. According to figure 6, the difference of lift coefficient between mesh level 3 which has 1362844 elements and mesh level 4 which has 2589254 elements accounts for 9.274percentage and the difference between mesh level 4 and 5 (4706688 elements) forms 4.780percentage. In addition, the difference value of level 5 and 6 (8228096) is 4.496. The last two figures show the small difference percentage (less than 5percentage), so the mesh level 5 that has element exceed 4700000 was chosen to be the simulation grid number.

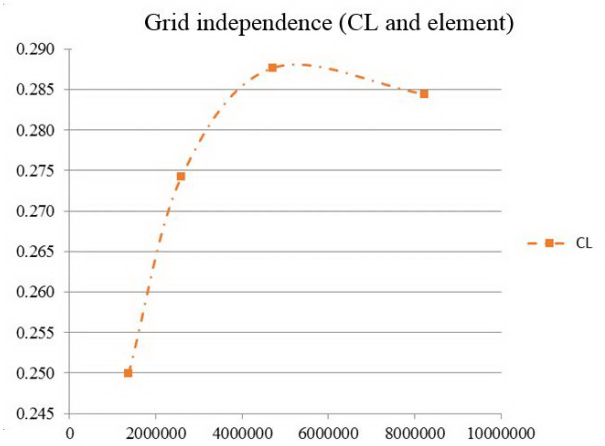


Figure 5: Grid independence.

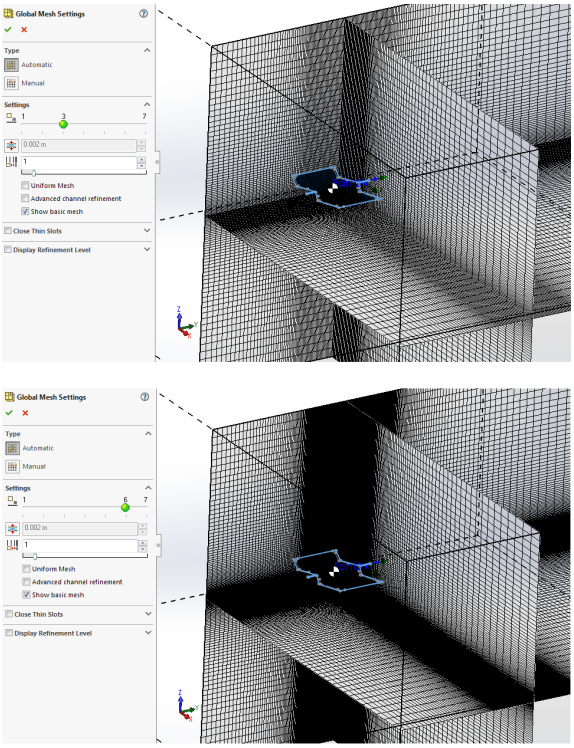


Figure 6: Compare mesh level between level 3 (left) and level 6 (right).

2.7 *Simulation of asymmetry wing*

The simplified folding wings were simulated in CFD program to compute rolling moment generated by asymmetry wing planform between two side. The wing was folded in 3 levels as shown in figure 7-9 respectively. The area of two side wing in each level were illustrated in table 3. Because of asymmetry of wing, they cannot be simulated by using symmetry domain to reduce the computer resources. Thereby, these three case of folding use full domain. The condition is, however, similar to the assigned condition for the plain wing such as the wind velocity of 15 m/s and 6-degree angle of attack.

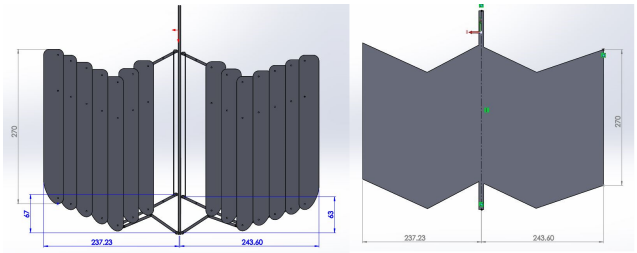


Figure 7: Folding wing level 1 (actual: left, simplified: right).

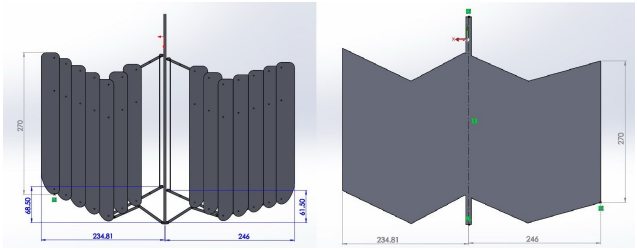


Figure 8: Folding wing level 2 (actual: left, simplified: right).

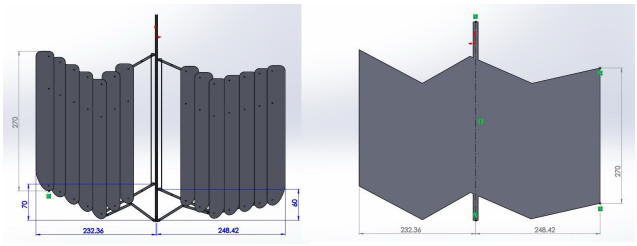


Figure 9: Folding wing level 3 (actual: left, simplified: right).

3 Results

3.1 Aircraft characteristic for plain wing

After set all of the condition in the simulation program and mesh the model at the chosen level, aircraft characteristics of the plain wing were determined by CFD approach to calculate all of the aerodynamic forces that acting on the wing at any angle of attack. This process is conducted to ensure the trim angle of attack of the wing and to find certain significant characteristics that will illustrate the stability and aerodynamic performance of this wing. Raw data collected from program result were transferred to aerodynamic force by their definition. Eventually, they were calculated to be their coefficients in the different angle of attack that was the characteristics of this wing as shown in table **??**. The characteristics is 2 groups. One is force coefficients including lift coefficient and drag coefficient, they represent by CL and CD. The other one is moment coefficient that calculate around center of gravity. They are pitching (Cm), rolling (Cl) and yawing (Cn) moment coefficient. There are graphs that plot between angle of attack and force coefficients in figure 10 and the moment coefficients in figure 11. Then, the lift curve slope was determined. This makes up approximately 0.057 1/deg. In addition, the trim angle was found when the lift force equal to weight that was 0.5 kg, so the trim angle of this wing is approximately 6 degrees in cruise condition. Lift over drag coefficient was calculated and plot in figure 12.

Parameters	Plain wing	Fold 1	Fold 2	Fold 3
Right span;m	.240	.237	.235	.232
Left span;m	.240	.244	.246	.248
Right pin;mm	65.00	67.00	68.50	70.00
Left pin;mm	65.00	63.00	61.50	60.00
Right area;m2	.06620	.06541	.06481	.06421
Left area;m2	.06620	.06699	.06758	.06817

Table 3: Lift coefficient at any mesh level and the different between two level.

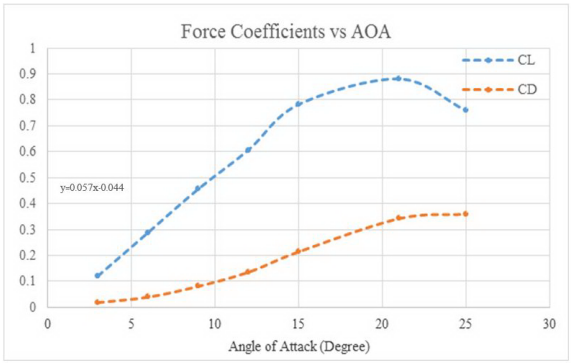


Figure 10: Force coefficient vs angle of attack.

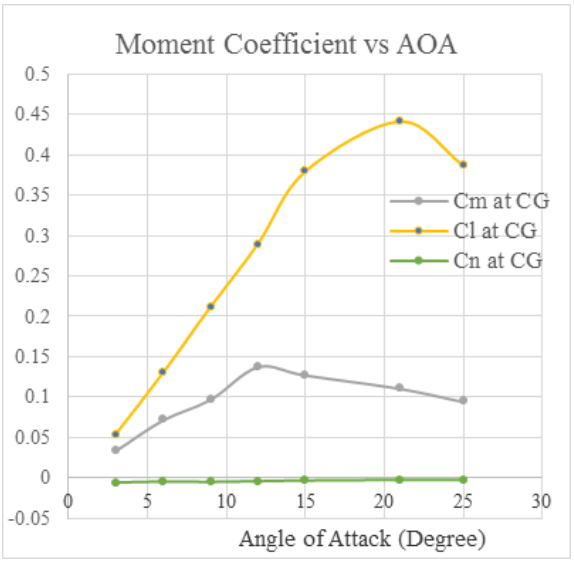


Figure 11: Moment coefficient vs angle of attack (left) and lift to drag ratio vs angle of attack (right).

Lift curve slope illustrated in figure 10 represents 0.057 (1/deg.) that can account for 3.264 (1/rad). This figure is lift curve slope that compute from 0 to 15 degree, so this may not be linear graph. The recalculated lift curve slope from 0 to 6 degree which should be linear represents 2.744 1/rad (0.0479 1/deg). There is a decrease in the slope, however, it is closer to the lift curve slope of the rectangular flat plate that have same as the aspect ratio [7]. If the slope of flat plate and the designed wing compare to the Helmholds lifting line theory,

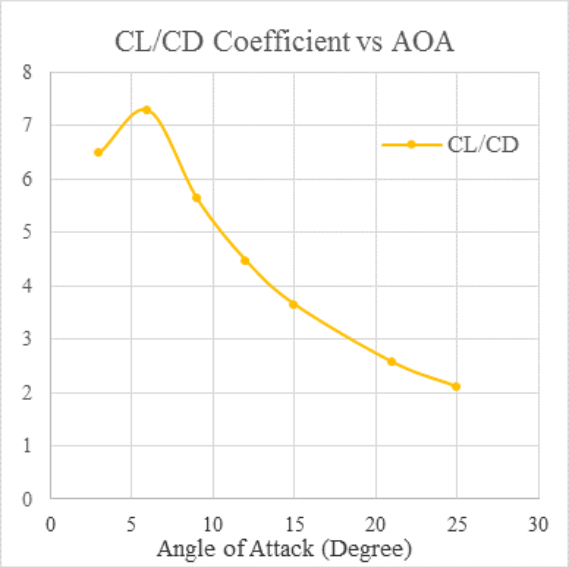


Figure 12: Moment coefficient vs angle of attack (left) and lift to drag ratio vs angle of attack (right).

the both errors shown in table 4 are insignificant difference. This may be because the equation is developed for low aspect ratio wings.

	Helmhold	Design wing	Flat plate
Lift curve slope	2.499	2.744	2.612
Error;percentage	-	9.817	4.543

Table 4: Lift curve slope.

Following step, the Oswalds efficiency factor (e) of the designed wing and the flat plate was computed to show the performance of the two wings from Prandtl's lifting line equation [6]. The figures illustrated that the design appears to be better performance than the rectangular flat plate. This may be the result of the strange shape that can produce the suitable flow.

(AoA 0-6deg)	CL(alpha) ;per deg	CL(alpha) ;per rad	Oswald factor
Designed wing	0.0479	2.744	0.820
Flatplate	0.0456	2.612	0.7532

Table 5: Oswalds efficiency factor.

3.2 Rolling moment generated from the folding wing

After folded the wing according to the models in section 2.6 and simulate them to calculate the moment that was created in each model, the both table 8 and figure 12 indicate the relationship between generated rolling moment (in term of rolling coefficient,) and the ratio of left and right wing area. It can be seen that the rolling moment increases when the differential area goes up. In the first state, it is symmetrical wing, so the coefficient of rolling moment is getting close to zero. Then the wing was folded, the area ratio changed. This results from an asymmetric lift force between two sides. The right wing has more area than the left, then the rolling moment was generated. When the different area increases in the second and third model, this results in a noticeable rise in the rolling moment.

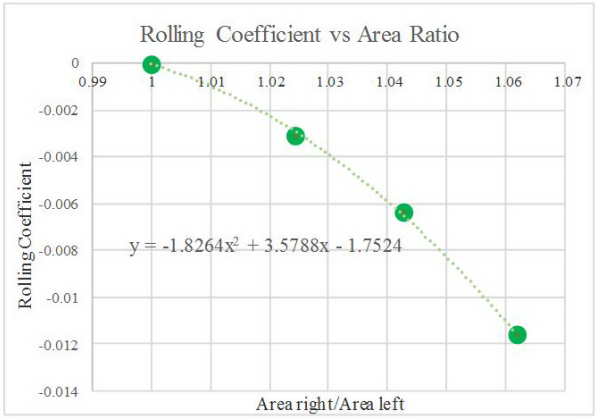


Figure 13: Relationship between rolling coefficient and area ratio.

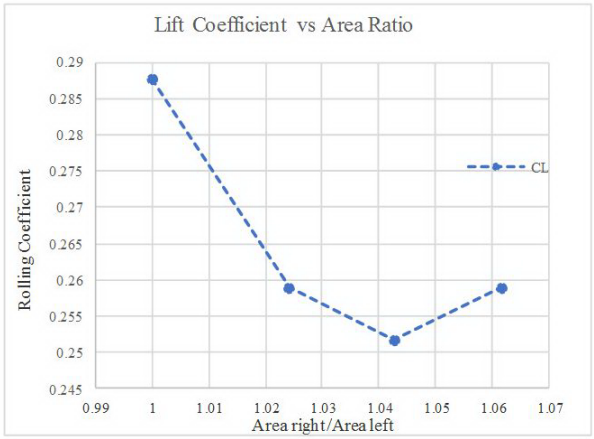


Figure 14: Relationship between CL and area ratio.

Furthermore, the relationship between lift coefficient and area ratio (figure 13) of this wing are in the range from around 0.285 to 0.25, so the required lift of the plane (0.274) is in this range. That mean this plane have enough lift to operate in rolling condition. Another relation between rolling coefficient and area ratio in figure 14 and equation on the graph meet the required rolling moment that was -0.00349 when the area ratio is 1.0279.

4 CONCLUSION

In conclusion, the study of this wing without folding is show the trim angle of attack and other significant aircraft characteristics. This will increase developers confidence to make the actual model of this aircraft. Another section about the folding wing indicates the positive result of the relation between the rolling coefficient and the area ratio of two side of wing. This convinces the developer that the folding aircraft is maneuverable by their structure without any rolling control surface.

ACKNOWLEDGEMENTS

This research could not acheived without the support from the Center of Innovative and Integrated of Mini and Micro Air Vehicle and other peoples in the Department of Aerospace Engineering, Faculty of Engineering, Kasetsart University. Authors would like to thank for the partial funding of registration fee for the conference from the Graduate School , Kasetsart University.

Figure 15: CiiMAV, Department of Aerospace Engineering, Kasetsart University

REFERENCES

[1] S.Kaewsutthi C.Thiptopas and A.Tohwae a yee. High performance propeller system for a multi-mission micro aerial vehicle. *International Journal of Micro Air Vehicles*, 5(3):179–191, 2013.

[2] J.M.Moschetta K.Chinwicharnam, D.Ariza and C.Thiptopas. Aerodynamic characteristics of a low aspect ratio wing and propeller interaction for a tilt-body mav. *International Journal of Micro Air Vehicles*, 5(4):245–260, 2013.

[3] M.J. Bamber. Aerodynamic rolling and yawing moments produced by floating wing-tip ailerons, as measured by the spinning balance. In *National advisory committee for aeronautics*, 1934.

[4] D.M. Rao and H. Va. Leading edge flap system for aircraft control augmentation. In *United States Patent, 4485992*, 1984.

[5] R.C. Montgomery D.L. Raney and L.L. Green. Flight control using distributed shape-change effector arrays. In *AIAA Paper*, 2000.

[6] P.G. Ifju. Flexible-wing-based micro air vehicles. In *WIT Transactions on State of the Art in Science and Engineering*, 2005.

[7] T.J. Mueller. Aerodynamic measurements at low reynolds numbers for fixed wing micro-air vehicles. In *Hessert Center for Aerospace Research, University of Notre Dame*, 1999.

Development of Vision Based Navigation for Micro Aerial Vehicles in Harsh Environment

Hailong Qin*, Yingcai Bi, F. Lin and Ben M. Chen
National University of Singapore, Singapore

ABSTRACT

Safe navigation and planning capabilities are of great importance to Micro Aerial Vehicles (MAVs) applications in different environments. MAVs are allowed to quickly access to the open sky or open area where far away from ground obstacles based on the onboard Global Positioning System (GPS) data. Except for these areas, MAVs become more demanding for missions in complex harsh environment, such as autonomous exploration, inspection and rescue tasks in disaster areas due to their maneuverability and low cost. To complete these missions, visual navigation for MAVs has been extensively studied because of the abundant 3D information and lightweight property.

In this article, a complete navigation system with a multi-heterogeneous sensor setup for visual sensing in harsh environment is proposed. Measurements of multiple RGBD visual sensors are utilized for localization in unknown harsh environment. After this, the measurement from visual sensors is fused with MAV onboard Inertial Measurement Unit (IMU) information through a preintegration approach to achieve autonomously takeoff, navigate and landing in unknown harsh environment. Extensive experiments have been conducted in both indoor and outdoor environments to evaluate the performance of the proposed system. Moreover, a preliminary fast navigation in challenge long corridor environment is also conducted to verify the robustness of the system.

1 INTRODUCTION

Due to small size and high manoeuvrability, MAVs have become a powerful and popular tool for rescue, surveillance and exploration. To perform such tasks, MAVs have to fly autonomously in unknown environments, which particularly depends on the results of localization and navigation. However, such operations are remarkably challenging in unstructured GPS-denied scenarios so that visual sensors, such as monocular and stereo cameras have become the most popular choices [1]. Furthermore, due to the stringent payload and power limit, using a high-power CPU or an advanced sensor set would not only significantly increases the power consumption and reduces the flight time, but also requires larger and more powerful motors and propellers, making the system more dangerous for civilian and defense applications. The characteristics of vision-based techniques has disadvantages of high sensitivity to illumination. Realizing localization and navigation for MAVs robustly and efficiently while employing auxiliary equipment as less as possible to save payload is still a critical problem and has been attracting increased attentions from researchers in control, robotics and vision communities.

To solve above issues, simultaneous localization and mapping (SLAM) has been popularly employed and comprehensively investigated for a robust navigation control loop of MAVs. Despite of its rapid progress and great achievements in recent years, SLAM still cannot meet some practical requirements. For instance, the classical SLAM is still a computational intensive process and hardly to be realized on portable computer to achieve real-time navigation [2] [3] [4]. In addition, due to the natural property of visual sensors, common vision-based SLAM can hardly work in low illumination environment.

Besides visual sensors, LIDAR is also a popular sensor used for SLAM problems and provides high frequency updating rate, wide-angle measurement and metric scale depth information [5]. Yet, LIDAR based localization algorithm is limited to the LIDAR sensing space constrain. For instance, using LIDAR to achieve 6 Degree of Freedom(DoF) pose estimation, it is usually required to employ a 3D LIDAR or a rotating 2D LIDAR. However, for both rotating 2D and 3D LIDAR, the point cloud distortion effect due to LIDAR motion still exists [6]. Thus additional sensors such as cameras are normally used for independent position estimation. Therefore, the LIDAR sensors are only largely deployed for dense mapping on Unmanned Ground Vehicles(UGVs) [7].

To operate MAVs successfully in harsh environment such as narrow featureless corridor, dark tunnel or firefighting scenario which filled with smoke, several challenges need to be addressed. First, the MAV should be highly manoeuvrable to fly in space limited environment with onboard sensors. Second, all the perception, planning and obstacle avoidance algorithms should run in real-time and be computational efficient for onboard processing. Third, multiple sensors in different directions should be carried and utilized so that the MAV can

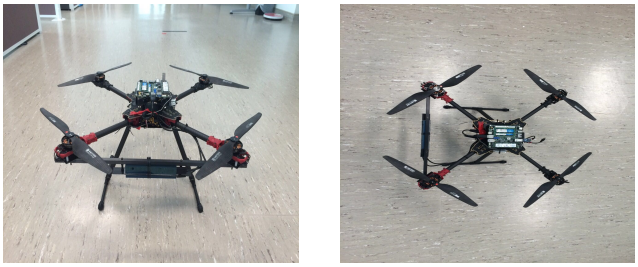
*Email address(es): qinhailong722@gmail.com

even navigate and avoid obstacles in potential dark environment.

In this article, the developed MAV prototype with onboard sensors is firstly presented. The developed prototype is a light weight but efficient platform with onboard powerful computer and multiple visual sensors from Intel named RealSense. The aim of the developed platform is to achieve autonomously takeoff, navigate and land in general daylight indoor and outdoor environment. The developed onboard RealSense based localization and reconstruction algorithm will be presented in the following sections.

2 SYSTEM CONFIGURATION

Our designed platform is a customized small-size and light-weight quadrotor platform. The general properties of quadrotor are fast vertical takeoff, general in air fly and vertical landing. Besides these, Quadrotor platfrom has certain payload to carry onboard sensors. For our designed platform, the Pixhawk developed by ETH is utilized as flight controller [8]. To achieve a stable and robust performance, the robust perfect tracking (RPT) algorithm is developed and implemented on the flight controller. The Intel developed x86 computer, NUC is installed for onboard high level computing such as visual perception, obstacle avoidance and planning. For onboard perception, the depth camera developed by Intel named Realsense is installed in front of MAV platform for visual odometry. Another same type depth camera is downward facing the ground for velocity estimation and precise height measurement. Based on these design requirements, our designed platform has a tip to tip length of 108cm. The designed platform is shown in Fig 1.



(a) MAV platform in front view (b) MAV platform in top down view

Figure 1: MAV platform in front view and top down view

The control system is divided into outer-loop using the RPT control concept and inner-loop with PID controller. The overall control structure can be described in following Fig. 2. Based on the generated position,velocity and acceleration reference from the mission control. The outer-loop controller generates the a_c. With a global-to-body transformation, the attitude references$(\phi_c, \theta_c, \psi_c)$ can be obtained. The detailed model formulation can be found in [9].

Since the visual odometry provides position measurement

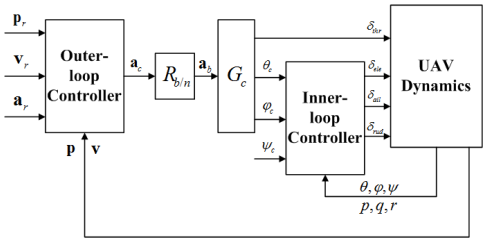


Figure 2: The dual loop control structure

with high frequency noise and low update rate, it can not be directly used as the input for control. Here, we adopt the Kalman filter in [9] to fuse the position measurement and provide a smooth MAV state estimation with a high update rate.

The Kalman filter framework can be described as a discrete-time linear system as following equation

$$\begin{aligned} x(k+1) &= Ax(k) + B(u(k) + w(k)) \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{1}$$

where x represent the state information. The input vector u equals to $(a_g(k) + w_g(k))$ which is the acceleration in local NED frame. w and v are input and measurement noises. With the designed filter, the system measurement updates at 50 HZ.

3 VISION BASED NAVIGATION FRAMEWORK

As discussed in previous sections, for the purpose of navigating MAVs in GPS-denied environment, additional perception sensors are mandatory for sensing the environment. Visual SLAM or visual odometry is the process of estimate the ego motion from continuous image sequence. The supplied abundant information and light weight of camera makes it a perfect choice for MAV navigation. However, neither monocular camera nor stereo camera can provide the depth information of the environment directly, which is critical to the perception and navigation of MAVs. Different from traditional cameras, our onboard Realsense camera is a infrared depth camera that able to measure the depth of pixel directly even in low illumination environment. Therefore, we can save the computational cost of onboard computer to the largest degree. Our proposed vision based navigation framework aims to solve the MAV perception issue in harsh environment such as dark tunnel.

Our depth estimation module contains two parts:(1) direct depth sensing (2) triangular depth sensing. For the direct depth sensing, the depth measurement is based on the output depth image. However, the depth information is not continuous and smooth enough for pixel based depth sensing. Therefore, a series of following processing modules are used to smooth and register available depth to image pixels.

1. **Median Filtering**
   The depth image captured by the RealSense is quite noisy. A fast, patch based median filter is utilized to

reduce the noise. The path has a size of 4 *by* 4. After this process, the depth measurement tends to be smooth with less noise.

2. **Depth Registration**
   The great advantage of the depth camera is that the depth can be directly captured and stored into depth image. Yet, the color image should be aligned with the depth image since there is a spatial difference between the color image and depth image. Through coordinate transformation, the color image can be registered with depth image.

3. **Pass Through and Noise Filter**
   In consideration of the outlier noise caused by the complicated environment, a statistical outlier remove algorithm is applied to remove the small isolated point blobs generated by the camera[10].



Figure 3: color image of forest



Figure 4: depth image of forest

Since not all the pixel depth can be referred from the direct depth, classical triangular depth sensing approach is required. The depth of the sparse visual features are based on the matched feature pair from consecutive valid image sequence. Concurrently, the odometry information can be obtained from multiple observations. During the whole navigation process, a sensing and control technique is needed to extract valuable information from outside environment for the purpose of controlling the MAV pose and localizing itself in an unknown environment. The onboard camera is not only light weighted but also give a comprehensive information in various environments. With the rapid development of monocular camera based visual SLAM, the MAVs equipped with onboard camera are able to provide spatial localization information from the visual odometry system [11]. However, due to the natural disadvantage of the monocular camera which converts a 3D world into a 2D image, the real scale of world MAV facing is lost. This drawback can be alleviated to some extent during landing since the MAV is trying to track a fixed target: either a feature based landmark or a suitable landing area.

Until now, several visual odometry techniques are available for MAVs' navigation and close the control loop. In [12], a realtime optical flow method implemented on monocular camera is presented. However, only the velocity information is estimated which can not give a full state information for control and the variation of light condition in outdoor environment can lead to bad performance. Andrew et al. first realize a realtime visual SLAM on a monocular camera in [13]. Yet this methodology also suffer the issue of depth lost from the beginning.

Different from other approaches, our proposed approach is a depth aware visual odometry without explicit loop closure. During the initialization, the scale and initial motion is obtained from the known depth based on the sparse feature optical flow. Followed by continuously depth and motion estimation from sparse feature matching and association. A local sparse feature with known depth is kept for feature matching.

The classical sparse feature based odometry using monocular camera has established a sophisticated motion estimation framework. Starting from feature detection, the depth (up to a scale) of the detected feature is continuously estimated from feature association. In addition, the camera pose is optimized according to the tracking error. To improve the accuracy and robustness of the tracking, the detected feature is inserted into a 3D map. The camera pose is further refined from matching with built map. The designed process is quite computational intensive. Therefore, we propose a multi-sensor fusion based visual odometry framework to improve the feature perdition accuracy and utilize the direct depth sensing to reduce computational load.

3.1 IMU Preintegration

The tightly-coupled approach is widely utilized in current vision based estimation. The motion between camera frames is precomputed through sensing information from IMU. After this, the IMU error term is computed and optimized with reprojection error to achieve a fast and highly accurate motion estimation. However, the optimization process requests the IMU information to be inserted into a precise timestamp.

Therefore, a dedicated IMU triggered camera capture system is highly demanding which most of commercial camera system does not achieve. In our proposed framework, we only implemented an IMU preintegration between each selected frame to speed up the feature association process. In this framework, we define the world reference frame as W and IMU frame as I. Thus, the rotation, velocity and position of IMU in world reference frame between time interval k is calculated as:

$$R_{WB}^{k+1} = R_{WB}^{k} Exp((\omega_B - b_g)\Delta t) \tag{2}$$

The velocity and position can also be calculated as following equations respectively:

$$v_{W,I}^{k+1} = v_{W,I}^{k} + g_W \Delta t + R_{WB}^{k}(a_I^k - b_a^k)\Delta t \tag{3}$$

$$p_{W,I}^{k+1} = p_{W,I}^{k} + v_{W,I}^{k}\Delta t + \frac{1}{2} g_W \Delta t^2 + \frac{1}{2} R_{WB}^{k}(a_I^k - b_a^k)\Delta t^2 \tag{4}$$

Where b_g and b_a are the varying biases of gyroscope and accelerometer respectively. ω and a are the measurement from IMU.

3.2 Visual Odometry

We define the camera frame as C. Thus, the 3D coordinate of a detected feature i in current frame k is defined as $X_C^{k,i}$. Therefore, if feature i can be detected in selected frame k and $k+1$, the rigid motion relationship can be expressed as

$$X_C^{k+1,i} = R X_C^{k,i} + T \tag{5}$$

The detected features and corresponding depth map for indoor and outdoor condition are shown in Fig. 5 and Fig. 6, where the green points stand for direct depth measurement and blue points stand for indirect depth measurement.

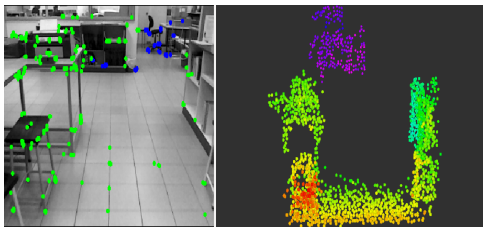


Figure 5: Visual features and depth map in indoor environment

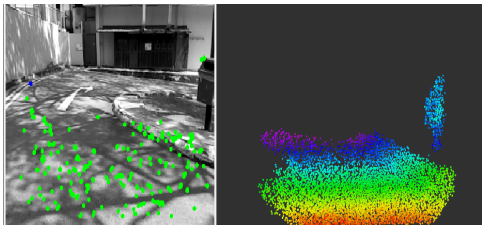


Figure 6: Visual features and depth map in outdoor environment

For the feature with known depth the above equation can be directly transformed into

$$(R_1 - \overline{x_i} R_3) X_C^{k,i} + T_1 - \overline{x_i} T_3 = 0 \tag{6}$$

$$(R_1 - \overline{y_i} R_3) X_C^{k,i} + T_1 - \overline{y_i} T_3 = 0 \tag{7}$$

Where $\overline{x_i}$ and $\overline{y_i}$ are the normalized coordinate by depth measurement z_i.

Using multiple detected features, a nonlinear function can be built using above equations

$$f(R,T) = d \to 0 \tag{8}$$

Moreover, the square error function can be established as

$$S = (f(R,T))^2 \tag{9}$$

Based on the derived error function, the Trust-Region-Reflective method (TRR) [14] is utilized to minimize the error and obtain the transformation matrix.

4 EXPERIMENTAL RESULTS

The proposed framework is verified in challenging wind gust, open outdoor and long corridor indoor environment to demonstrate the accuracy and robustness. Although we are not able to create a real harsh environment to demonstrate the performance of the designed framework, we claim that the designed experiments can test the system to a certain degree.

4.1 Hovering in wind gust

Wind gust is occasionally happened in harsh environment. Therefore, MAV with stable flight capability even in wind gust is highly demanded. The designed wind gust includes random wind with speed of 2 m/s, 4 m/s and 5 m/s. In this autonomous hovering experiment, our designed platform shows outstanding stability using proposed vision based navigation framework. The result is shown in Fig. 7. From the result we can see that the MAV is continuously resist wind to maintain the position and maintain the position error in a reasonable range.

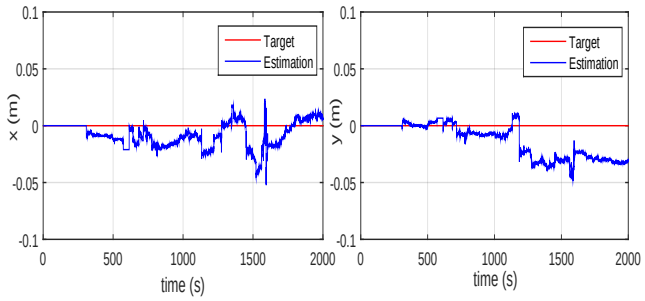


Figure 7: Horizontal motion estimation for MAV hovering in wind gust.

4.2 *Flying in outdoor environment*

A large outdoor close loop experiment is conducted to demonstrate that our design framework can help MAV to localize and reconstruct the surrounding environment. Since the ground truth is not available, the close loop path is designed to verify the accuracy of the framework. The detected feature and reconstructed dense environment is shown in Fig. 8 and Fig. 9.

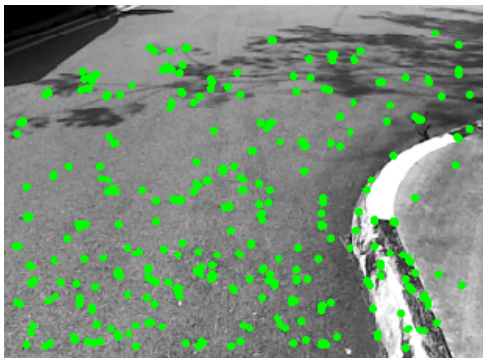


Figure 8: Detected feature in outdoor experiment.

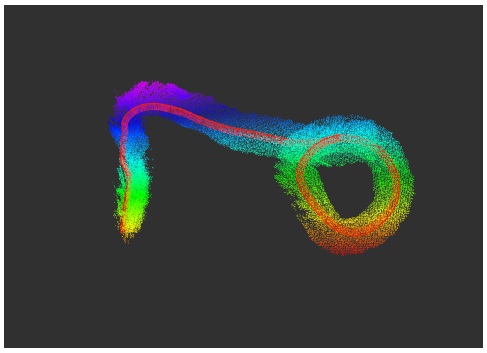


Figure 9: Reconstructed environment of outdoor environment(Estimated trajectory in red).

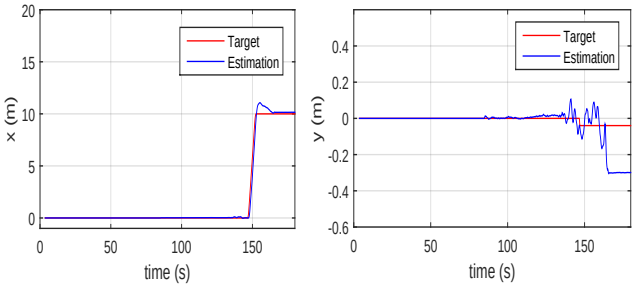


Figure 10: Horizontal motion estimation of high speed flight in long corridor.

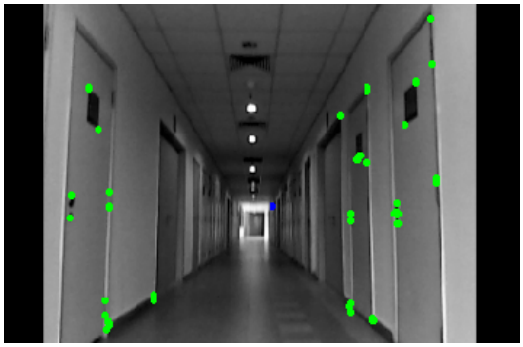


Figure 11: Detected features in long corridor.

Due to limited sensing range of RGBD camera we can see only see limited field of view as shown in Fig. 8. However, uniform distributed visual feature can still be detected to achieve motion estimation. In addition, the large scale circular motion also demonstrate the loop closure accuracy of our proposed framework as shown in Fig. 9.

4.3 *Flying long corridor environment*

Finally, a challenging autonomous flight in long corridor environment is conducted. Due to the limited sensing range of visual sensors, distinct visual feature is preferred to build robust feature alignment. However, this requirement is hard to achieve in homogeneous environment such as long corridor with texttureless wall. We demonstrate that our proposed framework can still achieve stable performance in this challenging environment with a average flight speed of 2.3 $\mathrm{m/s}$.

The motion estimation result is shown in Fig. 10. As we can see from results, MAV closely tracks the target reference even in high speed motion with reasonable overshoot. In addition, few feature can be detected in this environment as shown in Fig. 11. Yet, the low number infeature will not greatly affect the performance of our proposed framework.

In addition, a back and forth autonomous flight experiment is conducted to demonstrate the stability of the proposed framework in high speed motion. the motion estimation re-

sult and reconstructed dense environment is shown in Fig. 12 and Fig. 13.

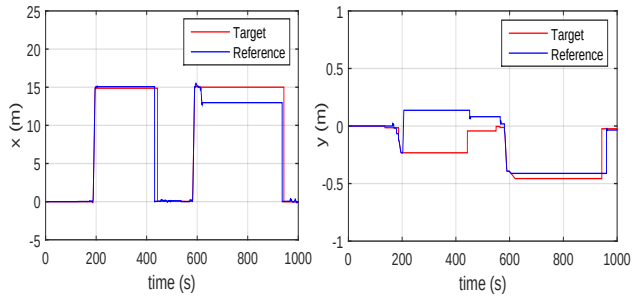


Figure 12: Horizontal motion estimation of high speed back and forth flight in long corridor.

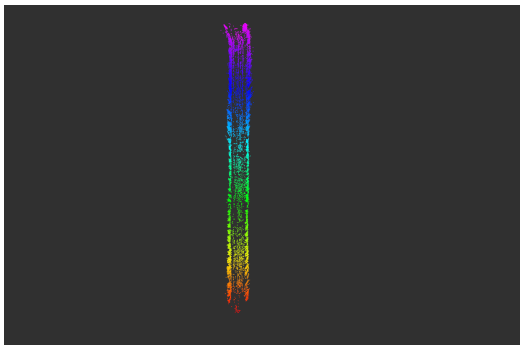


Figure 13: Reconstructed dense corridor environment.

Both the motion estimation with target tracking and clear reconstructed dense corridor verify the accuracy and robustness of our framework as wrong motion estimation will lead to corrupted reconstruction.

5 CONCLUSION

In this article, a complete RGBD camera based navigation framework for MAVs in harsh environments is presented. The vision estimated odometry is further optimized through an IMU preintegration approach. Finally, the realtime estimation is utilized as MAV localization information through a Kalman filter. We demonstrated the robustness of proposed approach in wind gust and efficiency in both indoor and outdoor GPS-denied environments. The experimental results clearly show the outstanding performance of our designed framework regardless of the harsh environments.

REFERENCES

[1] Lionel Heng, Dominik Honegger, Gim Hee Lee, Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer, and Marc Pollefeys. Autonomous visual mapping and exploration with a micro aerial vehicle. *Journal of Field Robotics*, 31(4):654–675, 2014.

[2] Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011.

[3] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *arXiv preprint arXiv:1607.02565*, 2016.

[4] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[5] Stefan Kohlbrecher, Oskar Von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 155–160. IEEE, 2011.

[6] Ji Zhang and Sanjiv Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast.

[7] Yihuan Zhang, Jun Wang, Xiaonian Wang, Chaocheng Li, and Liang Wang. A real-time curb detection and tracking method for ugvs by using a 3d-lidar sensor. In *Control Applications (CCA), 2015 IEEE Conference on*, pages 1020–1025. IEEE, 2015.

[8] Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer, and Marc Pollefeys. Pixhawk: A system for autonomous flight using onboard computer vision. In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 2992–2997. IEEE, 2011.

[9] Guowei Cai, Ben M Chen, and Tong Heng Lee. *Unmanned rotorcraft systems*. Springer Science & Business Media, 2011.

[10] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. Interpretation of urban scenes based on geometric features.

[11] Michael Blosch, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Vision based mav navigation in unknown and unstructured environments. In *Robotics and automation (ICRA), 2010 IEEE international conference on*, pages 21–28. IEEE, 2010.

[12] Volker Grabe, HH Bulthoff, and Paolo Robuffo Giordano. On-board velocity estimation and closed-loop control of a quadrotor uav based on optical flow. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 491–497. IEEE, 2012.

[13] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052–1067, 2007.

[14] Ya-xiang Yuan. A review of trust region algorithms for optimization.

Efficient Global Indoor Localization for Micro Aerial Vehicles

V. Strobel*[1], R. Meertens[2], and G.C.H.E. de Croon†[2]

[1]IRIDIA, Université Libre de Bruxelles, Brussels, Belgium
[2]Micro Air Vehicle lab, Control and Simulation, Faculty of Aerospace Engineering, Delft University of Technology, the Netherlands

Abstract

Indoor localization for autonomous micro aerial vehicles (MAVs) requires specific localization techniques, since GPS is usually not available. We present an onboard computer vision approach that estimates 2D positions of an MAV in real-time. The global localization system does not suffer from error accumulation over time and uses a k-Nearest Neighbors (k-NN) algorithm combined with a particle filter to predict positions based on textons—small image patches. The performance of the approach can be predicted by an evaluation technique that compares environments and identifies critical areas within them. In flight tests, the algorithm had a localization accuracy of approx. 0.6 m in a 5 m×5 m area at a runtime of 32 ms on board of an MAV. Its computational effort is scalable to different platforms, trading off speed and accuracy.

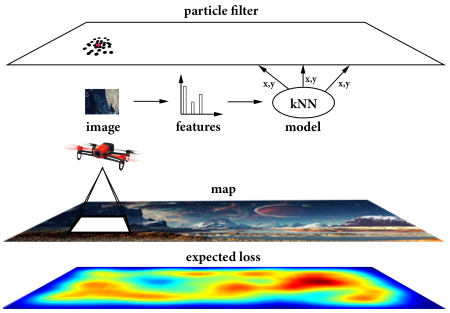


Figure 1: The figure illustrates the presented system from a high-level perspective. The feature vector—the texton histogram—that is extracted from the current camera image is forwarded to a machine learning model that uses a k-Nearest Neighbors algorithm to output k x, y-position estimates. These estimates are passed to a particle filter, which filters position estimates over time and outputs a final position estimate (red point). The expected loss shows regions in the map where a lower localization accuracy is expected.

1 Introduction

Accurate onboard localization is a key challenge for micro aerial vehicles (MAV). In confined spaces, specific localization algorithms are essential, since the Global Positioning System (GPS) is usually not available. While light-weight MAVs could be employed in various indoor tasks, they cannot fall back on standard localization approaches due to their limited payload and processing power. To address this issue, this paper presents an efficient indoor localization technique (Figure 1). Our contribution is a machine learning-based indoor localization system that runs onboard of an MAV paving the way to an autonomous system. In the presented approach, computational power is shifted to an offline training phase to achieve high speed during live operation. In contrast to visual SLAM frameworks, this project considers scenarios in which the environment is known beforehand or can even be actively modified. The approach is based on the occurrence of textons, which are small characteristic image patches. With textons as image features and a k-Nearest Neighbors (k-NN) algorithm, we obtain 2D positions in real-time within a known indoor environment. A particle filter was developed that handles the estimates of the k-NN algorithm and resolves positional ambiguities. We consider settings in which the MAV moves at an approximately constant height, such that the estimation of height is not necessary. In contrast to existing approaches that use *active* sensors, the developed approach only uses a *passive* monocular downward-looking camera. While carrying active sensors, such as laser range finders, is too demanding for a light-weight MAV, onboard cameras can typically be attached. Additionally, we developed a technique for evaluating the suitability of a given environment for the presented algorithm. It identifies critical areas and assigns a global loss value to an environment. This allows for comparing different potential maps and identifying regions with low expected localization accuracy. The developed global localization system does not suffer from error accumulation over time. Onboard processing helps to reduce errors and delays introduced by wireless communication, and ensures a high versatility on the way to an autonomous system. We evaluated the the approach in flight experiments.

The remainder of this paper is structured as follows. Sec-

*vstrobel@ulb.ac.be
†g.c.h.e.decroon@tudelft.nl

tion 2 surveys existing indoor localization approaches. In Section 3, the developed texton-based approach is presented and its components, the k-NN algorithm and the particle filter, are introduced. Section 4 describes the setup and results of the flight experiments. The results are discussed in Section 5 and we draw conclusions in Section 6.

2 Related Work

While a wide range of methods for indoor localization exists, we only consider methods in this section that use the same technical and conceptual setup—localization with a monocular camera.

One distinguishes two types of robot localization: local techniques and global techniques [14]. Local techniques need an initial reference point and estimate coordinates based on the change in position over time. Once they lost track, the position can typically not be recovered. The approaches also suffer from "drift" since errors are accumulating over time. Global techniques are more powerful and do not need an initial reference point. They can recover when temporarily losing track and address the *kidnapped robot problem*, in which a robot is carried to an arbitrary location [13].

2.1 Optical Flow

Optical flow algorithms estimate the apparent motion between successive images. The most popular optical flow methods are gradient based approaches and keypoint-based methods [4]. Optical flow methods belong to the class of *local* localization techniques and most approaches are computationally rather complex [4].

2.2 Fiducial Markers

Fiducial markers have been used for UAV localization and landing [12, 21]. The markers encode information by the spatial arrangement of black and white or colored image patches. Their corners can be used for estimating the camera pose at a high frequency. An advantage of fiducial markers is their widespread use, leading to technically mature and open-source libraries. A drawback of the approach is that motion blur, which frequently occurs during flight, can hinder the detection of markers [1]. Furthermore, partial occlusion of the markers through objects or shadows break the detection. Another downside is that markers might be considered as visually unpleasant and may not fit into a product or environmental design [5].

2.3 Homography Determination & Keypoint Matching

A standard approach for estimating camera pose is detecting and describing keypoints of the current view and a reference image [22], using algorithms such as Scale-invariant feature transform (SIFT) [19], followed by finding a homography—a perspective transformation—between both keypoint sets. A keypoint is a salient image location described by a feature vector. Depending on the algorithm, it is invariant to different viewing angles and scaling.

This *homography-based* approach is employed in frameworks for visual Simultaneous Localization and Mapping (SLAM) but the pipeline of feature detection, description, matching, and pose estimation is computationally complex [15]. While the approach has been employed for global localization for UAVs, the required processing power is still too high for small MAVs [7].

2.4 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a specialized machine learning method for image processing [18]. The supervised method has outperformed other approaches in many computer vision challenges [10]. While their training is usually time-consuming, predictions with CNNs often takes only few milliseconds, shifting computational effort from the test phase to the training phase. CNNs have been used as robust alternative for keypoint detection and description if images were perturbed [10] but needed more computation time than SIFT. In recent work, Kendall, Grimes, and Cipolla present a framework for regressing camera positions based on CNNs [15]. The approach is rather robust to different lighting settings, motion blur, and varying camera intrinsics. The approach predicts positions on a modern desktop computer in short time.

2.5 Texton-based Methods

Textons [24] are small characteristic image patches; their frequency in an image can be used as image feature vector. A texton histogram is obtained by extracting patches from an image and comparing them to all textons in a "texton dictionary". The frequency of the most similar texton is then incremented in the histogram.

Texton histograms are flexible image features and their extraction requires little processing time, which makes them suitable for MAV on-board algorithms. The approach allows for adjusting the computational effort by modifying the amount of extracted image patches, resulting in a trade-off between accuracy and execution frequency [8].

De Croon et al. [7] use textons as features to distinguish between three height classes of the MAV during flight. Using a nearest neighbor classifier, their approach achieves a height classification accuracy of approximately 78 % on a hold-out test set. This enables a flapping-wing MAV to roughly hold its height during an experiment. In another work, De Croon et al. [9] introduce the *appearance variation cue*, which is based on textons, for estimating the proximity to objects [9]. Using this method, the MAV can avoid obstacles in a $5\,m \times 5\,m$ office space.

3 Methods

The pseudo code in Algorithm 1 shows a high-level overview of the parts of the framework. Details are given in the following subsections.

Algorithm 1 High-level texton framework

```
t ← 0
X_0 ← INIT_PARTICLES
while true do
    t ← t + 1
    I_t ← RECEIVE_IMG_FROM_CAMERA
    H_t ← GET_TEXTON_HISTOGRAM(I_t)
    z_t ← k-NN(H_t)
    X_t ← PARTICLE_FILTER(X_{t-1}, z_t)
    x_t, y_t ← MAP_ESTIMATE(X_t)
end
```

3.1 Hardware and Software

We used the quadcopter *Parrot Bebop Drone* as a prototype for all our tests. The developed approach uses the bottom camera only, which has a resolution of 640 × 480 pixels with a frequency of 30 frames per second.

3.2 Dataset Generation

A main idea of the presented method is to shift computational effort to a pre-flight phase. Since the MAV will be used in a fixed environment, the results of these pre-calculations can be employed during the actual flight phase. Supervised machine learning methods need a training set to find a mapping from features to target values. In this first step, the goal is to label images with the physical x, y-position of the UAV at the time of taking the image.

One possible way to create the data set is to align the images with high-precision position estimates from a motion tracking system, which yields high-quality training sets. Major disadvantages of the approach are that motion tracking systems are usually expensive and time-consuming to move to different environments.

As an alternative, we sought a low-budget and more flexible solution. Out of the presented approaches in Section 2, the homography-based approach (Section 2.3) promises the highest flexibility with a good accuracy but also requires the most processing time. Since fast processing time is not relevant during the pre-flight phase, the approach is well-suited for the problem. The required image dataset can be obtained by using images gathered during manual flight or by recording images with a hand-held camera. To get a hyperspatial image of the scene for creating a map, the images from the dataset have to be stitched together. With certain software packages the images can be orthrectified by estimating the most probable viewing angle based on the set of all images. However, since a downward-looking camera is attached to the UAV, most images will be roughly aligned with the z-axis, given slow flight [3]. For the stitching process, we used the freeware software Microsoft Image Composite Editor (ICE) [20]. Keypoints of the current image and the map image are detected and described using the SIFT algorithm. This is followed by a matching process, that identifies corresponding keypoints between both images. These matches allow for finding a homography between both images. For determining the x, y-position of the current image, its center is projected on the reference image using the homography matrix.

3.3 Texton Dictionary Generation

For learning a suitable texton dictionary for an environment, image patches were clustered. The resulting cluster centers—the prototypes of the clustering result—are the textons [25]. The clustering was performed with a Kohonen network [16]. The first 100 images of each dataset were used to generate the dictionary. From each image, 1 000 randomly selected image patches of size $w \times h = 6 \times 6$ pixels were extracted, yielding $N = 100\,000$ image patches in total that were clustered. For our approach, we also used the color channels U and V from the camera to obtain color textons.

3.4 Histogram Extraction

The images from the preliminary dataset are converted to the final training set that consists of texton histograms and x, y-values. To extract histograms in the *full sampling* setting, a small window—or kernel—is convolved across the width and height of an image and patches are extracted from all positions. Each patch is compared with all textons in the dictionary and is labeled with the nearest match based on Euclidean distance. The histogram is normalized by dividing the number of cases in each bin by the total number of extracted patches, to yield the relative frequency of each texton.

The convolution is a time-consuming step, since all possible combinations of width and height are considered: $(640 - w + 1) \cdot (480 - h + 1) = 301\,625$ samples are extracted. To speed up the time requirements of the histogram extraction step, the kernel can be applied only to randomly sampled image position instead [8]. This sampling step speeds up the creation of the histograms and permits a trade-off between speed and accuracy. The random sampling step introduces random effects into the approach. Therefore, for generating the training dataset, no random sampling was used to obtain high-quality feature vectors.

3.5 k-Nearest Neighbors (k-NN) algorithm

The k-Nearest Neighbors (k-NN) algorithm is the "machine learning-core" of the developed algorithm. Taking a texton histogram as input, the algorithm measures the Euclidean distance of this histogram to all histograms in the training dataset and outputs the k most similar training histograms and the corresponding x, y-positions.

While the k-NN algorithm is one of the simplest machine learning algorithms, it offers several advantages [17]: it is non-parametric, allowing for the modeling of arbitrary distributions. Its capability to output multiple predictions enables neat integration with the developed particle filter. Additionally, k-NN regression often outperforms more sophisticated algorithms [6]. A frequent point of criticism is its increasing computational complexity with an increasing size of the training dataset. While the used training datasets consisted of fewer than 1000 images, resulting in short prediction times (see also Figure 6), time complexity can be reduced by stor-

ing and searching the training examples in an efficient manner, for example, with tree structures [2].

3.6 Filtering

Our approach uses a filtering method that is able to capture *multimodal distributions*. Given an adequate measurement model, a general Bayesian filter can simultaneously maintain multiple possible locations and resolve the ambiguity as soon as one location can be favored. In this case, the predictions of the k neighbors can be directly fed into the filter without averaging them first. However, a general Bayesian filter is computationally intractable. Therefore, a variant based on random sampling was used: the particle filter. While its computational complexity is still high compared to a Kalman filter, one can modify the amount of particles to trade off speed and accuracy and adapt the computational payload to the used processor.

The weighted particles are a discrete approximation of the probability density function (*pdf*) of the state vector (x, y-position of the MAV). Estimating the filtered position of the MAV can be described as $p(X_t \mid Z_t)$, where X_t is the state vector at time t and $Z_t = \mathbf{z}_1, ..., \mathbf{z}_t$ are all outputs of the k-NN algorithm up to time t, with each $\mathbf{z}_i$ representing the k x, y-outputs of the algorithm at time i.

The used particle filter is initialized with particles at random x, y-positions. To incorporate the measurement noise for each of the k estimates from the k-NN algorithm, we developed a two-dimensional Gaussian Mixture Model (GMM) as measurement model. The GMM is parameterized by the variances $\Sigma^{[j]}, j \in \{1, \ldots, k\}$ that are dependent on the rank j of the prediction of the k-NN algorithm (for example, $j = 2$ is the second nearest neighbor). The variance matrix $\Sigma^{[j]}$ specifies the variances of the deviations in x-direction and y-direction and the correlation ρ between the deviations. The values for $\Sigma^{[j]}$ were determined by calculating the variance-covariance matrix for the difference between the ground truth T from the motion tracking system and the predictions P_j of the k-NN algorithm: $\Sigma^{[j]} := \mathrm{Var}(T - P_j)$.

The used *motion model* is solely based on Gaussian process noise and does not consider velocity estimates, headings, or control inputs. Its mean and variance are dependent on the expected velocity of the MAV. We used the forward difference $T_t - T_{t-1}$ to estimate the average movement and its variance-covariance matrix Σ_{process} between timesteps t and $t-1$.

In the following pseudo code of the developed particle filter (Algorithm 2), $\mathcal{X}$ is the list of particles, f the two-dimensional Gaussian probability density function, $z_t^{[i]}$ the ith neighbor from the kNN prediction, $x_t^{[m]}$ the mth particle at time t, and $w_t^{[m]}$ its corresponding weight. The "resampling wheel" [23] performs the importance resampling step.

With the GMM, the information of all k neighbors can be used, yielding a possibly multimodal distribution. While a multimodal distribution allows for keeping track of several possible positions, certain subsystems—for example a control loop—often need *one* point estimate. Using a weighted average of the particles would again introduce the problem that it could fall into a low density region (an unlikely position). Instead, we used a maximum a posteriori (MAP) estimate, as described by Driessen and Boers [11]. The estimation of *uncertainty* was modeled using the spread of the particles—as expressed by their variance in x-direction and y-direction.

Algorithm 2 Particle filter update

```
1: procedure PARTICLE_FILTER(X_{t-1}, z_t)
2:     ▷ Initialize particle list
3:     X_temp := ∅
4:     for m = 1 to M do
5:         ▷ Add random process noise (motion model)
6:         x_t^[m] ← x_t^[m] + N(0, Σ_process)
7:         ▷ Iterate over k-NN preds (measurement model)
8:         w ← 0
9:         for i = 1 to k do
10:            ▷ Gaussian Mixture Model
11:            w ← w + f(z_t^[i]; x_t^[m], Σ_measurement^[i])
12:        X_temp := X_temp ∪ (x_t^[m], w)
13:    ▷ Importance resampling
14:    X_t ← RESAMPLING_WHEEL(X_temp)
15:    return X_t
```

3.7 Map evaluation

The performance of the developed method depends on the environment: a texture-rich environment without repeating patterns will be better suited than a texture-poor environment. To assess if the algorithm will work in a given environment, we propose an evaluation scheme that compares different environments and areas within an environment. This scheme assigns a global fitness value or global loss value to a "map"—expressed as dataset $\mathcal{D}$ consisting of N texton histograms h_i and corresponding x, y-coordinates $\text{pos}_i = (x_i, y_i)$. The fitness value is intended to be proportional to the accuracy that can be expected when using this dataset as training set for the developed localization algorithm. The scheme allows for inspecting the dataset and detecting regions within the map that are responsible for the overall fitness value.

The idea behind the global loss function L is that histograms h_i and h_j in closeby areas should be similar and the similarity should decrease with increasing distance of the corresponding x, y-coordinates pos_i and pos_j. Therefore, the approach is based on the difference between *actual* and *ideal* texton histogram similarities in a dataset. The ideal texton similarity distribution is modeled as a two-dimensional Gaussian distribution around each x, y-position in the dataset. Using this idea, a histogram is compared to all others by comparing expected similarities to actual similarities. This results in a loss value per sample of the dataset (local loss). Applying the algorithm to each sample in the dataset yields the global loss of a dataset.

The method uses the cosine similarity $CS(h_i, h_j) = \frac{h_i^T h_j}{||h_i|| \, ||h_j||}$ to compare histograms. The cosine similarity has the convenient property that its values are bounded between -1 and 1. In the present case, since the elements

of the histograms are non-negative, it is even bounded between 0 and 1. Let $f(x;\mu,\sigma) = e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ describe the non-normalized one-dimensional Gaussian probability density function. Since we assume that the ideal similarity in x-position is independent of the y-position, the ideal two-dimensional similarity function $d_e(\text{pos}_i, \text{pos}_j; \Sigma)$ can be modeled as the product of the respective one-dimensional function f:$d_e(\text{pos}_i, \text{pos}_j; \Sigma) = f(x_i; x_j, \sigma_x) \cdot f(y_i; y_j, \sigma_y$ This function is also bounded between 0 and 1, which makes the functions d_e and CS—ideal similarity and actual similarity—easily comparable. In summary, we propose the following global loss function (L) for evaluating a given dataset ($\mathcal{D}$): $L(\mathcal{D}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} CS(h_i, h_j) - f(x_i; x_j, \sigma_x) \cdot f(y_i; y_j, \sigma_y)$ The simple difference—in contrast to least absolute deviations or least square errors—ensures that similarities that are *less* similar than the ideal similarity *reduce* the loss. Therefore, a high variation in texture is always seen as "positive". The variances σ_x and σ_y specify the dimension of the region, where similar histograms are desired. The lower their value, the more focused the ideal similarity will be, requiring a high texture variety for getting a low loss value. A high value might overestimate the suitability of a dataset. While the approach is relatively robust to the choice of the parameter values, we still need to find a heuristic for suitable values.

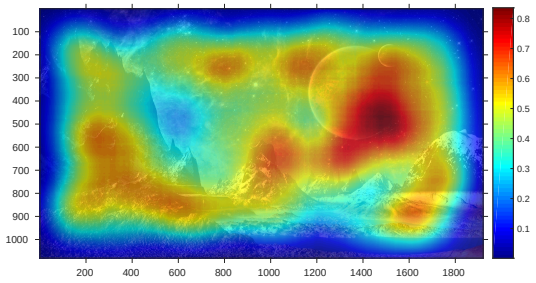


Figure 2: The figure shows the loss of a map: the regions that did not follow the ideal similarity pattern are displayed in red. For the visualization, the loss values per sample in the dataset were smoothed with a Gaussian filter. This assigns a loss value to each x, y-position of the map.

4 Analysis

In the experiments, the MAV was guided along flight plans using the motion tracking systen. If not otherwise stated, we used the following default values for the parameters in our framework:

parameter	value
samples in the histogram extraction step	400
textons in the dictionary	20
particles of the particle filter	50
histograms / images in the training set	800
histograms / images in the test set	415
neighbors in the k-NN algorithm	5

Map-dependent texton dictionaries were used and created by conducting an initial flight over the respective maps.

4.1 Baseline: Homography-based Approach

To find a baseline for our approach and to provide a homography-based training set, we used the homography-based approach to estimate x, y-coordinates in the same environment and based on the same images as the texton-based framework. The required hyperspatial image (Figure 3) of the environment was stitched together using 800 images and the software Microsoft ICE.



Figure 3: The created map (size: approximately 5×5 meters) that was stitched together using 800 images.

We estimated the x, y-coordinates of the 415 test images using the homography-based approach and compared the predictions to the ground truth. The predictions were not filtered. The results can be found in the following table.

	x-position	y-position
Error in cm	31	59
STD in cm	68	77

4.2 Training Set based on Motion Tracking System

In this experiment, the position estimates were calculated on board of the MAV using the texton-based approach with the particle filter. The Euclidean distances between the estimates of the motion tracking system and the texton-based approach were measured in x-direction and y-direction.

The training dataset was composed of 800 texton histograms with corresponding x, y-coordinates that were obtained from the motion tracking system. The images were recorded in a 5×5 meter area at a height of approximately one meter in a time span of one hour before the experiment to keep environmental factors roughly the same.

The results can be found in the following table. They are based on 415 images, which corresponds to a flight time of approximately 35 seconds.

	x-position	y-position
Error in cm	61	59
STD in cm	39	39

4.3 Training Set based on Homography-finding Method

In this experiment, the training dataset was created by estimating the x, y-positions of the 800 training images using the homography-finding method from the previous section and the same hyperspatial image. Apart from that, the settings are the same as in the previous experiment.

	x-position	y-position
Error in cm	54	97
STD in cm	41	61

4.4 Triggered Landing

For the triggered landing experiment, the MAV was guided along random flight paths, which covered a 5×5 meter area; during navigation, the MAV was programmed to land as soon as its position estimates were in a "landing zone": an x, y-position with a specified radius r. A safety criterion was introduced such that the landing is only performed if the standard deviations of the particles in x-direction and y-direction are below thresholds θ_x and θ_y. We set the parameters to $\theta_x = \theta_y = 60\,\text{cm}$. The x, y-coordinate of the circle was specified in the flight plan; the radius was set to $r = 60\,\text{cm}$. We performed six triggered landings; after each landing, the x, y-center of the zone was randomly set to another position in the map. For the texton framework, the same training set as in Experiment 4.2 was used.

Four out of six landings were correctly performed in the landing area. The distances of the two outliers were 14 cm and 18 cm, measured as distance to the circumference of the landing area.

4.5 Speed versus Accuracy Trade-Off

Adapting the frequency of the main loop of the developed approach to make it suitable for different platforms with varying processing power is one of its core parts. Figures 4 and 5 show the speed versus accuracy trade-off as a function of the used particles and of the used samples in the histogram extraction step, respectively. As a reference, the frequency using *full sampling* in the histogram extraction step was 0.1 Hz. The above stated *default* values were used for the *ceteris paribus* assumption, when varying the parameters. While the bottom camera of the Parrot Bebop Drone has a frequency of 30 Hz, the Paparazzi software currently only receives the images with a frequency of 12.5 Hz. Therefore, the baseline for the conducted experiment—the maximum achievable frequency—is 12.5 Hz. Figure 6 illustrates the frequency as a function of the used histograms in the k-NN algorithm. After having received the image, the processing time of the presented algorithm using the *default* parameter values is 32 ms, which includes the histogram extraction (16 ms) as well as the k-NN predictions, the filtering and the output of the best x, y-coordinate (16 ms).

5 Discussion

The flight tests show initial evidence for the real-world suitability of the method, which yields slightly less accurate results than the unfiltered homography-finding method. While we did not test the frequency of the homography-based approach on board of an MAV, on a desktop computer, it took 200 ms per image. Therefore, the developed algorithm runs at a much higher frequency. The triggered landing (Experiment 4.4) showed good accuracy: while most landings were triggered inside the landing zone, two out of the six landings were outliers. However, their distance to the landing area were rather small, with an average distance of 16 cm.

The experiments show that with an increasing accuracy of the approach, the frequency of the algorithm decreases. However, the errors reach a plateau after which no large improvements can be expected at the lower end of parameter ranges. By optimizing the parameters, one can obtain localization errors below 50 cm with the developed approach.

While we compared the settings of different parameters, there are no generally optimal parameters for the presented framework: setting the number of textons, the number of images patches, or the number of neighbors is dependent on the environment and the size of the training dataset. The parameters have to be adapted to the particular environment.

The accuracy of our global localization technique could be further improved by combining it with a local technique. To this end, odometry estimates using optical flow or the inclusion of data from the inertial measurement unit (IMU) could be suitable.

Our current implementation assumes constant height up to few centimeters and only small rotations of the MAV. While a quadroter can move in every direction without performing yaw movements, other MAVs or the use of the front camera for obstacle avoidance could require them. The inclusions of images of arbitrary yaw movements into the dataset would inflate its size to a great extent. This could lead to a deterioration of the accuracy and increase the time-complexity of the k-NN algorithm. Instead, a "derotation" of the camera image based on IMU data could be performed to align it with the underlying images of the dataset.

6 Conclusion

This paper presented an approach for lightweight indoor localization of MAVs. We pursued an onboard design to foster real-world use. The conducted experiments underline the applicability of the system. Promising results were obtained for position estimates and accurate landing in the indoor environment. An important step in the approach is to shift computational effort to a pre-flight phase. This provides the advantages of sophisticated algorithms, without affecting performance during flight. The approach can trade off speed with accuracy to use it on a wide range of models. The map evaluation technique allows for predicting and improving the quality of the approach.

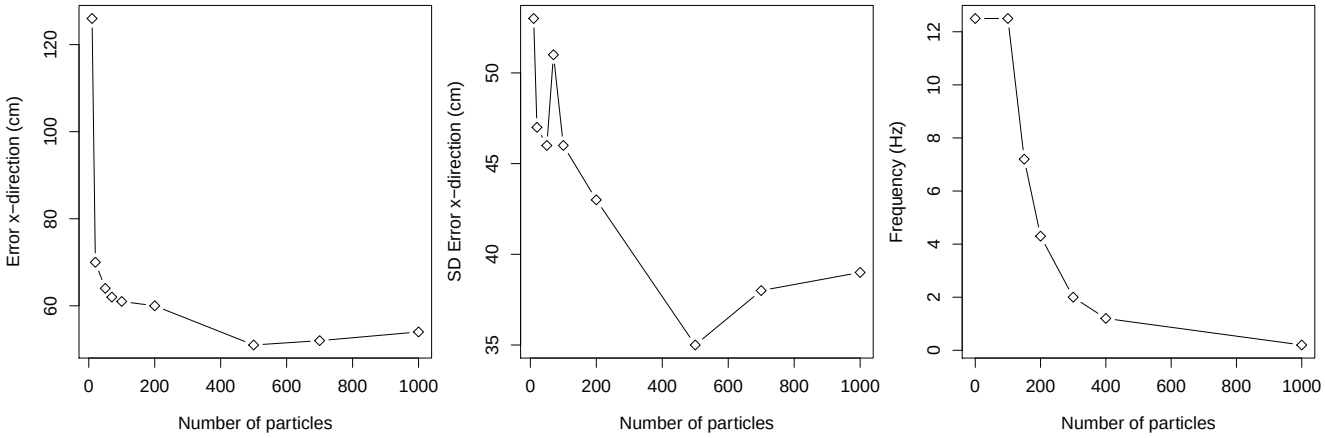


Figure 4: Speed versus accuracy trade-off in x-direction as a function of the number of used particles.

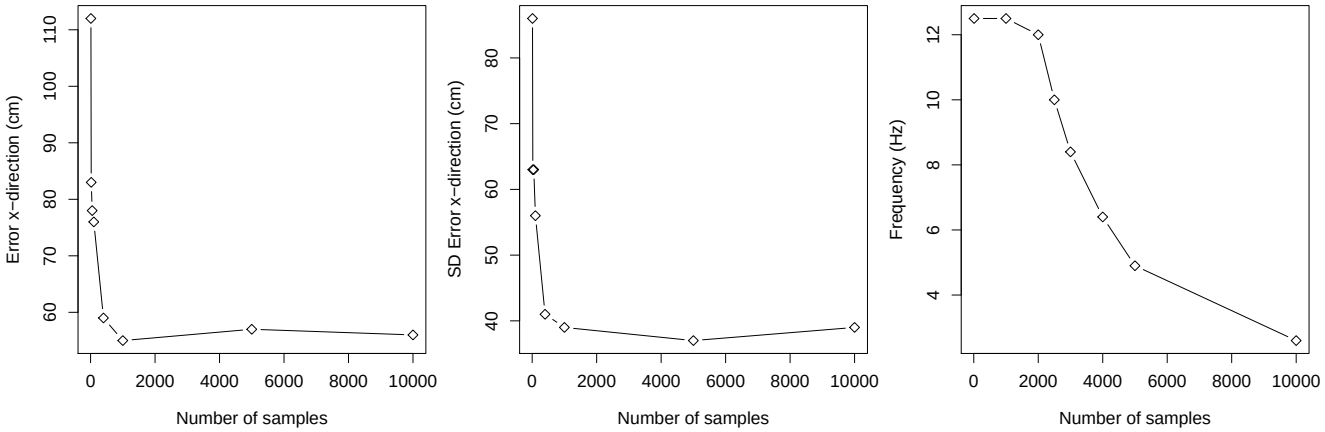


Figure 5: Speed versus accuracy trade-off in x-direction as a function of the number of used samples in the histogram extraction step.

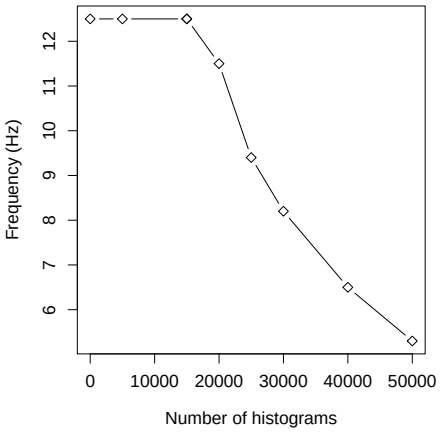


Figure 6: Frequency of the main loop as a function of the number of histograms in the training set.

REFERENCES

[1] Eman R. AlBasiouny, Amany Sarhan, and T. Medhat. "Mean-shift-FAST algorithm to handle motion-blur with tracking fiducial markers". In: *Proc. IEEE Conference on Computer Engineering & Systems (ICCES 2015)*. IEEE. 2015, pp. 286–292.

[2] Nitin Bhatia et al. "Survey of nearest neighbor techniques". In: *arXiv preprint arXiv:1007.0085* (2010).

[3] Michael Blösch et al. "Vision based MAV navigation in unknown and unstructured environments". In: *Proc. IEEE International Conference on Robotics and Automation (ICRA 2010)*. IEEE. 2010, pp. 21–28.

[4] Haiyang Chao, Yu Gu, and Marcello Napolitano. "A survey of optical flow techniques for UAV navigation applications". In: *Proc. International Conference on Unmanned Aircraft Systems (ICUAS 2013)*. IEEE. 2013, pp. 710–716.

[5] Hung-Kuo Chu et al. "Halftone QR codes". In: *ACM Transactions on Graphics (TOG)* 32.6 (2013), p. 217.

[6] *CPSC 340: Machine Learning and Data Mining*. URL: https://www.cs.ubc.ca/~schmidtm/Courses/340-F15/L7.pdf (visited on 08/10/2016).

[7] G.C.H.E. De Croon et al. "Design, aerodynamics, and vision-based control of the DelFly". In: *International Journal of Micro Air Vehicles* 1.2 (2009), pp. 71–97.

[8] G.C.H.E. De Croon et al. "Sub-sampling: Real-time vision for Micro Air Vehicles". In: *Robotics and Autonomous Systems* 60.2 (2012), pp. 167–181.

[9] G.C.H.E. De Croon et al. "The appearance variation cue for obstacle avoidance". In: *IEEE Transactions on Robotics* 28.2 (2012), pp. 529–534.

[10] Alexey Dosovitskiy et al. "Discriminative unsupervised feature learning with convolutional neural networks". In: *Advances in Neural Information Processing Systems*. 2014, pp. 766–774.

[11] Hans Driessen and Yvo Boers. "MAP estimation in particle filter tracking". In: *2008 IET Seminar on Target Tracking and Data Fusion: Algorithms and Applications*. IET. 2008, pp. 41–45.

[12] Daniel Eberli et al. "Vision based position control for MAVs using one single circular landmark". In: *Journal of Intelligent & Robotic Systems* 61.1-4 (2011), pp. 495–512.

[13] Sean P. Engelson and Drew V. McDermott. "Error correction in mobile robot map learning". In: *Proc. IEEE International Conference on Robotics and Automation (ICRA 1992)*. IEEE. 1992, pp. 2555–2560.

[14] Dieter Fox et al. "Monte Carlo localization: Efficient position estimation for mobile robots". In: *Proc. National Conference on Artificial Intelligence/Annual Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI 1999)*. American Association for Artificial Intelligence, 1999, pp. 343–349.

[15] Alex Kendall, Matthew Grimes, and Roberto Cipolla. "PoseNet: A convolutional network for real-time 6-DOF camera relocalization". In: *Proc. IEEE International Conference on Computer Vision (ICCV 2015)*. 2015, pp. 2938–2946.

[16] Teuvo Kohonen. "The self-organizing map". In: *Proc. IEEE* 78.9 (1990), pp. 1464–1480.

[17] Mirosław Kordos, Marcin Blachnik, and Dawid Strzempa. "Do we need whatever more than k-NN?" In: *Proc. International Conference on Artificial Intelligence and Soft Computing (ICAISC 2010)*. Springer. 2010, pp. 414–421.

[18] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proc. IEEE* 86.11 (1998), pp. 2278–2324.

[19] David G. Lowe. "Object recognition from local scale-invariant features". In: *Proc. IEEE International Conference on Computer Vision (ICCV 1999)*. Vol. 2. IEEE. 1999, pp. 1150–1157.

[20] *Microsoft Image Composite Editor*. URL: http://research.microsoft.com/en-us/um/redmond/projects/ice/ (visited on 08/02/2016).

[21] Parrot. *CES 2015 : Parrot Bebop Dance Choreography*. 2015. URL: https://www.youtube.com/watch?v=A_3UifFb45Y (visited on 10/15/2015).

[22] Stephen Se, David Lowe, and Jim Little. "Global localization using distinctive visual features". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*. Vol. 1. IEEE. 2002, pp. 226–231.

[23] S. Thrun. *Artificial intelligence for robotics*. URL: https://www.udacity.com/course/artificial-intelligence-for-robotics--cs373 (visited on 03/10/2016).

[24] Manik Varma and Andrew Zisserman. "A statistical approach to texture classification from single images". In: *International Journal of Computer Vision* 62.1-2 (2005), pp. 61–81.

[25] Manik Varma and Andrew Zisserman. "Texture classification: Are filter banks necessary?" In: *Proc. IEEE Computer Society Sonference on Computer Vision and Pattern Recognition (CVPR 2003)*. Vol. 2. IEEE. 2003, pp. II–691.

3D Reconstruction of Complex Structures with Online Profiling and Adaptive Viewpoint Sampling

Abdullah Abduldayem, Dongming Gan, Lakmal Seneviratne, Tarek Taha*
Khalifa University, Abu Dhabi, United Arab Emirates

ABSTRACT

A modified Next Best View (NBV) approach is presented to improve the 3D reconstruction of complex symmetric structures. Two new stages are introduced to the NBV approach; A profiling stage quickly scans the structure of interest and builds a rough model called the "profile". The profile is relatively sparse but captures the major geometric features of the structure. A symmetry detection stage then determines major lines of symmetry in the profile and labels points of interest. If a point exists in known space but its mirror image lies in unknown space, the mirrored point becomes a point of interest.

The reconstruction is performed by a sensor mounted on an Unmanned Aerial Vehicle by using a utility function that incorporates the detected symmetry points. We compare the proposed method with the classical information gain utility function in terms of coverage completeness, total iterations, and travel distance.

1 INTRODUCTION

Once restricted to controlled industrial environments, robots have become increasingly autonomous in the past decade. One task being delegated to robots is the automatic 3D reconstruction of objects and structures where there is a need for high density reconstructions that ensure complete coverage of the structure. This technology has applications in inspection, model scanning and in preservation of archaeological artefacts in digital form. Performing the inspection task manually is time-consuming with no guarantees on reconstruction density, coverage completeness or repeatability.

In some cases, a model or blueprint of the structure is provided which can be used to pre-compute paths for the robot to follow as it reconstructs the structure [1, 2]. However, these models are not available or provided in many cases.

To overcome this problem, the robot iteratively constructs a map as it navigates around the object of interest. One such method is the so-called **Next Best View (NBV)** approach [3, 4, 5] which attempts to determine where to place the sensor next in order to obtain the most new information about the structure.

*Email: {abdullah.dayem, dongming.gan, lakmal.seneviratne, tarek.taha}@kustar.ac.ae

The basic NBV methodology works as follows: Starting from an initial position, the robot senses its surroundings and creates an initial map of the structure. The robot then generates a set of candidate viewpoints to move to next, evaluates each of these viewpoints, and finally moves to the view that maximizes the information gained about the structure. A utility function is usually used to evaluate these viewpoints by weighing factors such as information gain and effort required to reach that viewpoint. This process is repeated over and over, selecting new viewpoints at each iteration until the structure is fully modelled, or a termination criteria has been met.

2 BACKGROUND

Model-less based approaches incrementally build the map as they explore. Typically, these make use of a Next Best View (NBV) approach as described in the introduction. The method is often broken down into stages, with the two main components being Viewpoint Sampling and Viewpoint Evaluation.

2.1 Viewpoint Sampling

Some techniques that have been explored to generate candidate viewpoints are frontier approaches, Rapidly Exploring Trees (RRT) and discretized state space approaches.

A **frontier** is a boundary between explored and unexplored space [6]. The frontier is determined by constructing an occupancy grid which marks whether a cell is free, occupied or unexplored. Unexplored cells which are adjacent to free cells are frontier cells, and the chain of adjacent frontier cells is a frontier.

Frontier methods perform well indoors and in constrained areas. In these environments, the frontier exists only at the end of a corridor or start of a room. However in an outdoor setting, the sky represents a large unexplored space and, if there are no obstructions, the frontier can potentially reach the horizon and become enormous. Bircher et al. [5] shows that frontier approaches stall in outdoors environments, with most of this time spent on evaluating the multitude of viewpoints.

A related technique for reconstruction is the **boundary search** method [7]. Rather than finding the edge of explored *space*, it locates the edge of the explored *object*. Poses near the boundary are sampled, and the quality of the poses can be improved by estimating the shape of the unseen surface at the boundary [7]. A similar method locates "barely visible" surfaces whose normals exceed a defined visibility angle [8].

Rapidly-exploring Random Trees (RRT) sample viewpoints randomly and connects these viewpoints in a tree-like structure, expanding throughout the explored space. Each branch represents a set of viewpoints to be visited in sequence, and the branch which maps the most unmapped space is selected. These trees can be grown using RRT or the RRT* variant [9].

Rather than executing an entire branch at once, Bircher et al. [5] determines the best branch and executes only its first edge. The remainder of the branch is used to initialize a new tree, preserving the original path while allowing for new paths to grow as the world map is updated. This method prefers to yaw in place to minimize cost penalty while the frontier method always moves to edges of known space. In Bircher's tests, RRT performed similar to frontier in indoor environments but vastly outperforms it outdoors. This is because the number of voxels in the frontier representation grows exponentially with size so it takes longer evaluate all the possible frontiers.

The **discretized state space** —also known as state lattice, configuration space or discretized joint space— takes each degree of freedom (DOF) and divides its range into a number of discrete steps. The possible poses that an n-DOF robot can take are thus visualized as points lying in a discretized n-dimensional space. While this technique is usually applied to manipulators with fixed joint range [10], it can also be used on a mobile robot if an estimated volume for the workspace can be obtained. Those working with UGVs may use 2-D position and yaw while UAVs may use 3-D position and yaw [11]. Additional DOFs may be added if a pan-tilt camera is used to alter the pitch of the camera as well.

With each additional DOF, the number of points in the discretized space increases exponentially. Similarly, large scenarios and fine resolution increase the number of viewpoints drastically, making it unsuitable for large-scale high precision applications. In these cases, a method must be used to limit the selection before evaluating the next best view.

2.2 *Utility Functions*

A common strategy is to simply move to the closest viewpoint. This has been found to reduce movement time, processing time and effort but it results in reduced information gained per viewpoint [12]. Traditionally the robot would simply move to the closest frontier, but a variety of utility functions have been employed to achieve improved results [3].

A utility function is any function that attempts to rank the sampled viewpoints. Factors included in utility functions may include distance to viewpoint, view angle with respect to surface normal [7], and distance to hazards [13].

3 Proposed Method

Rather than attempting to tackle the NBV problem blindly, we propose a quick profiling stage. Before applying NBV, an unmanned aerial vehicle (UAV) scans the structure to obtain a rough initial model of the structure. The initial model will henceforth be referred to as the **profile**, as it shows the general shape of the structure and captures its significant geometric features. The process of obtaining the profile is called **profiling**.

After the profiling step, the main steps of the standard NBV method are executed as shown in Figure 1. The method samples a finite number of viewpoints, evaluates them and determines the next best viewpoint. The UAV then moves to the selected viewpoint and updates the map and structure representations. The termination condition is evaluated, and if the criteria is not satisfied, the process is repeated.

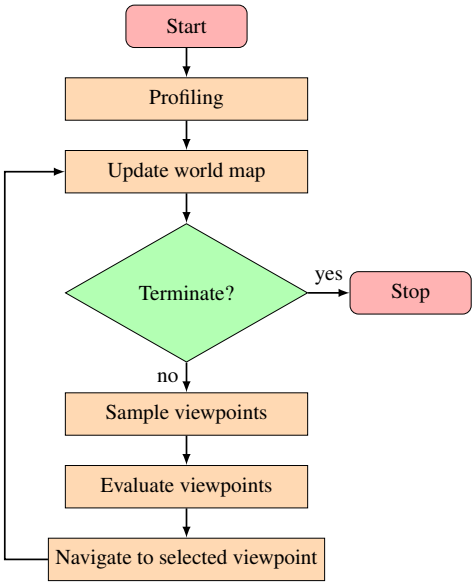


Figure 1: The basic components of the proposed NBV method

3.1 *Structure Representation*

As new point data is acquired by the sensors, two representations of the structure are updated: a point cloud and an occupancy grid. The occupancy grid divides space into a grid and stores a probability of occupancy in each cell. A probability of 1 means the cell is fully occupied, 0 is completely free and 0.5 is unknown as it is equally likely to be occupied or free. By using the notion of entropy from information theory, the amount of new information remaining in the scene can be computed. The entropy e_i of the cell i with occupancy probability p_i can be found by:

$$e_i = (p_i) \log (p_i) + (1 - pi) \log (1 - pi) \qquad (1)$$

The graph of e_i has a peak at $p_i = 0.5$, meaning unknown cells have the most information while completely free and completely occupied cells have no new information.

The occupancy grid is unsuitable for reconstructing the final structure due to its large spacial resolution. Instead, the final mesh of the structure is generated from a point cloud,

which is denser and more precise. Since the point cloud does not differentiate between free and unexplored space, it is necessary to make use of both occupancy grids and point clouds.

3.2 Profiling

The profiling stage initializes both the occupancy grid and the point cloud to guide further exploration.

To capture the profile, we have opted to use laser range data. While it is possible to obtain the profile using visual data, these tend to have poorer distance estimation at longer ranges and require multiple viewpoints to establish spatial positioning of an observed point. Visual sensors also rely on having a feature-rich environment to obtain features, so a reflective or plain surface would be captured with a relatively low point density.

On the other hand, range data has the benefit of collecting reliable 3D points from a single reading. LIDARs can take readings in nearly 360 degrees at distances up to 100 meters. The combined angular and distance capabilities allow the system to gather data about a large structure quickly.

Laser spatial resolution degrades with distance due to increasing distance between the radial rays, so point density will be lower at farther regions of the structure. Some areas may also be occluded, so it is necessary to explore the structure up close with a higher density sensor.

3.2.1 Adaptive Circular Profiling

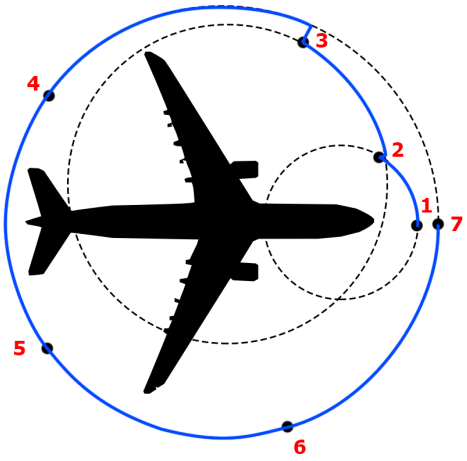


Figure 2: Example of profiling from top view. Dotted lines represent virtual cylinders that encapsulate all point data seen until that time instance. At each of numbered dot, the UAV scans vertically, updates the profile, and recomputes the radius and center of the bounding cylinder. The UAV then travels to the next waypoint and repeats.

We propose an adaptive method of obtaining a profile by moving in a circle around the structure whose radius expands to encapsulate the structure. The vehicle travels around a circle that fully encapsulates the points observed thus far. As the vehicle travels around the circle, the profile is updated and the centroid and bounding radius of the cylinder is adjusted. The procedure is illustrated in Figure 2 and described more formally in Algorithm 1.

Algorithm 1 Profiling: Adaptive Circular

Input:

- N — Number of viewpoints around structure for profiling
- d — Minimum distance for obstacle avoidance

Output:

- r_{occ} — an occupancy grid
- r_{cloud} — a point cloud

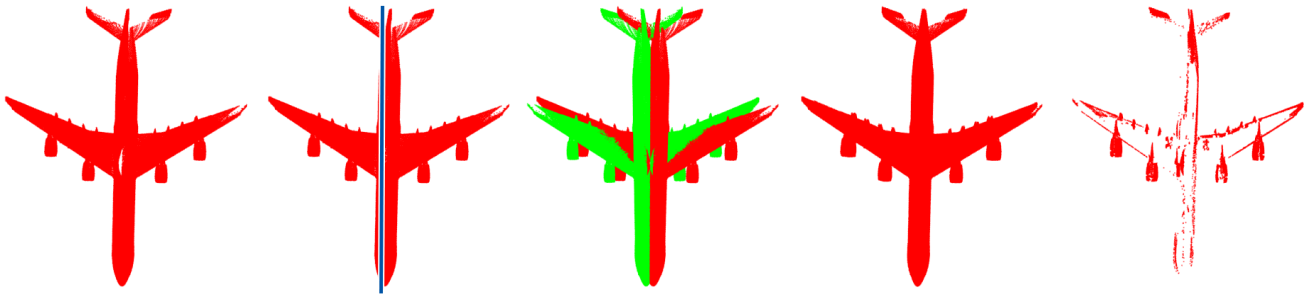
$\theta_s \leftarrow$ Angle between the global x axis and the vector from the UAV's starting position to the center of the workspace
$\theta \leftarrow \theta_s$
$\phi \leftarrow 2\pi/N$
while $(\theta - \theta_s) < 2\pi$ **do**
    Scan by moving vertically between z_{min} and z_{max}
    UpdateProfileMaps(r_{occ}, r_{cloud})
    $proj \leftarrow$ ProjectOntoXYPlane(r_{cloud})
    $c \leftarrow$ ComputeCentroid($proj$)
    $r \leftarrow$ GetBoundingRadius($proj$, c)
    $r \leftarrow r + d$
    MoveToCircleCircumference(c, r, θ)
    $\theta \leftarrow \theta + \phi$
    MoveAlongCircleCircumference(c, r, θ)
end while

3.2.2 Symmetric Prediction

The profile is likely to contain holes due to occlusion or limited sensor range. Many structures have symmetry, whether it is rotational, translational, intrinsic, extrinsic or otherwise. We can extract additional information about the structure by exploiting this property. In our application, we focus on reflectional symmetry.

In order to detect the line of symmetry, the following approach was taken:

1. Compute keypoints in the profile point cloud using Fast Point Feature Histograms (FPFH) [14]. By focusing on these keypoints, the number of points under consideration is reduced significantly.
2. Compute features for each keypoint using a variant of the Scale Invariant Feature Transform (SIFT) that operates on 3D point clouds. This variant has been implemented by Michael Dixon for the Point Cloud Library [15].



(a) (b) (c) (d) (e)

Figure 3: The symmetry-detection process for prediction. **(a)** The profile obtained by the simulated UAV. Some gaps exist in the model. **(b)** The detected line of symmetry shown in blue. **(c)** The profile is mirrored across the line of symmetry and shown in green. **(d)** The mirrored image is aligned with the original using ICP and merged to create the final prediction. **(e)** The additional points obtained from the symmetry process. These are visualized by subtracting the point cloud in (d) from (a)

3. Match each keypoint with the keypoint that has the most similar features (ie. Euclidean distance between features is minimum)
4. Fit a plane between each pair of points
5. Determine the plane of symmetry by performing mean shift clustering on all plane parameters computed in the previous step. Seen in Figure 3(b).
6. Create a copy of the profile point cloud and reflect it across the line of symmetry, as shown in Figure 3(c). This copy is called the **prediction**.
7. The predicted point cloud can be corrected by applying Iterative Closest Point (ICP) [16] between it and the profile, as shown in Figure 3(d).

Once we have obtained the predicted symmetric point cloud, we use it to modify the occupancy grid. Cells that are unknown but have a predicted point within them are updated to an occupancy value of 0.6. This update indicates that we have some confidence that the cell is occupied, but not as confident as a cell that has been directly observed.

3.3 Viewpoint Sampling

As there are an infinite number of poses to consider, attempting to determine the absolute best possible position in continuous space is intractable [12]. Instead, only a few selected viewpoints are sampled. To generate those viewpoints, we use a technique called **constant grid viewpoint sampling (CGVS)**.

The CGVS approach samples a few points in the vehicle's nearby vicinity in both linear and anglar space, as shown in Figure 4. While this method is not globally optimal, it ensures viewpoints in the nearby vicinity are sampled first to reduce travel cost.

This method can be generalized so that the sampling distance can vary. This may be done by multiplying linear distances with a scale factor of σ^{α}, where $\sigma > 1$ and $\alpha \geq 0$. This scaling approach extends CGVS and can be performed dynamically, giving rise to **adaptive grid viewpoint sampling (AGVS)**. The set of sampled viewpoints is given by Equation 2 with respect to the vehicle's local frame.

If the value of σ is fixed, the value of α must vary in order to change the scaling. Scaling occurs when the method fails to obtain sufficient entropy reduction after a given number of iterations. The criteria for "sufficient entropy reduction" is explained in Section 3.5. If Equation 6 is satisfied for $N = 3$ and $\Delta E_{v,threshold} = 0.005$, then α is incremented by one, otherwise it is reset to zero.

Finally, a viewpoint is considered **invalid and discarded** if it touches an occupied or unknown cell, collides with the structure or lies outside the workspace limits. If the method is unable to generate any valid viewpoints, the NBV process terminates.

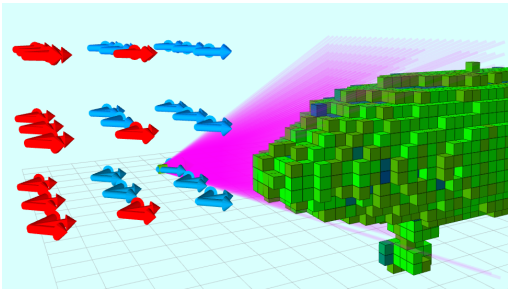


Figure 4: Constant grid viewpoint sampling method. Blue arrows represent valid viewpoints while red ones are rejected due to collision with the structure. Magenta lines represent the sensor's viewpoint at the selected viewpoint. Green voxels represent occupied space, while blue voxels show unknown space.

$$V_{CGVS} = \left\{ \begin{bmatrix} i \times \Delta d \times \sigma^{\alpha} \\ j \times \Delta d \times \sigma^{\alpha} \\ k \times \Delta d \times \sigma^{\alpha} \\ m \times \Delta\theta \end{bmatrix} \middle| \begin{array}{c} i,j,k,m \in \{-1,0,1\}, \\ \{i,j,k,m\} \neq \{0,0,0,0\} \end{array} \right\} \tag{2}$$

The first three elements of V_{CGVS} represent the spacial increment (x,y,z), while the last element represents the angular increment (yaw). The values of Δd and $\Delta\theta$ are in meters and radians, respectively. In our experiments, these values are set to $\Delta d = 1.0$m and $\Delta\theta = \pi/4$rad as that creates some overlap between viewpoints while providing the opportunity to discover new information. The value of σ is a fixed scale factor, while α is the number of times to apply the scale. We have used values of $\sigma = 1.5$ and $\alpha \in [0, 5]$ so the total scale factor varies from 1 to 7.594.

3.4 Viewpoint Evaluation

To determine which of the sampled viewpoints is the next best view, we select the viewpoint that maximizes a utility function. The utility function operates on a section of the point cloud and/or occupancy grid that is visible at that particular viewpoint.

3.4.1 Utility Function

The Information Gain (IG) metric is often used in literature [5, 3, 17], and will thus be used as a benchmark. It measures the total entropy (see Equation 1) of cells in a given viewpoint.

To determine which cells are visible from a given viewpoint, rays are cast according to the sensor's field of view. The ray passes through free or unknown cells, and stops once it collides with the first occupied cell in its path. The entropy of the cells along and at the endpoint of the ray trajectory is computed as in Equation 3

The total information H_k in a given viewpoint k which is able to see the set of cells V is given by:

$$H_k = \sum_{i \in V} e_i \tag{3}$$

In the proposed method, we make use of Equation 3 using our modified occupancy grid with predicted voxels.

3.4.2 Constraints

In the case where profiling is not performed, there is a large amount of unknown information surrounding the structure. Left on its own, the vehicle would prefer to select viewpoints with purely unknown cells and would move away from the structure. To prevent this, two restrictions are implemented.

First, viewpoints that do not observe at least 1 occupied cell are rejected. This encourages the vehicle to continue to observe the structure and overlap with previous observations.

Second, the structure lies within a bounded workspace. Any measurement outside those bounds is regarded as having zero utility so that even if the first restriction is removed, the vehicle would stay within the given bounds.

3.5 Termination

No consensus has been reached for a suitable termination condition as it varies between applications, and is also difficult to quantify the desired criteria. Ideally, the process should terminate once the system has achieved coverage completeness, but it is difficult to compute coverage without the original model for reference. The work in [18] terminates if the percentage difference in total entropy reduction has fallen below a threshold, while [12] terminates if no view has an IG above a certain threshold.

Both these methods have variable results. If the total number of unknown cells is very large, percentage difference in entropy reduction will be very small. This can happen if the size of the workspace increases, the occupancy cell size is reduced or many unknown spaces remain after the profiling stage. Similarly, it is possible to be temporarily trapped in local minima, so it may not be fair to terminate if IG reduction momentarily stagnates.

3.5.1 Entropy Change Per Viewable Voxel

To tackle the shortcomings of the previous termination conditions, we propose a termination condition based on the entropy reduction normalized by the maximum number of cells that can be observed from a given view. This gives average entropy reduction per cell and is given in Equations 4 and 5:

$$\Delta E_v = \frac{E_{n-1} - E_n}{C_{view}} \tag{4}$$

$$C_{view} = V_{frustum}/V_{cell} \tag{5}$$

where ΔE_v is the entropy change per cell, E_n is the total entropy at iteration n, C_{view} is the number of viewable cells, V_{cell} is the volume of a cell, $V_{frustum}$ is the volume of the view frustum which accounts for the camera's field of view.

In addition to counting the number of complete cells within a viewpoint, the value of C_{view} also includes partial voxels at the boundaries of the frustum. Sensor readings and raytracing can pass through these cells on the edge of the frustum, so it is necessary to count them as well.

We consider the NBV process has terminated if the condition in Equation 6 is true for more than N consecutive iterations. The value of N is selected so that the method does not terminate prematurely if it is stuck in a local minimum. The magnitude of ΔE_v is used rather than the raw value to account for loss of information. Entropy may increase if an obstacle is introduced or removed, or during raytracing with large occupancy cells from a variety of angles.

$$|\Delta E_v| < \Delta E_{v,threshold} \tag{6}$$

By computing the average entropy change in a single voxel, it is possible to have a single threshold (or tight range of thresholds) across a variety of scenario sizes and occupancy cell resolutions.

4 Experiments

4.1 Environment

Simulation experiments were performed on an Alienware-X51-R2 desktop (Intel Core i7-4790 @ 8 x 3.60 GHz, 15.6 GB RAM, no GPU acceleration). The NBV framework was implemented on Ubuntu 14.04 using the Robot Operating System (ROS-indigo) and simulated with Gazebo. The occupancy grid was represented with the OctoMap library [19] while point cloud data was processed using the Point Cloud Library (PCL) [15].

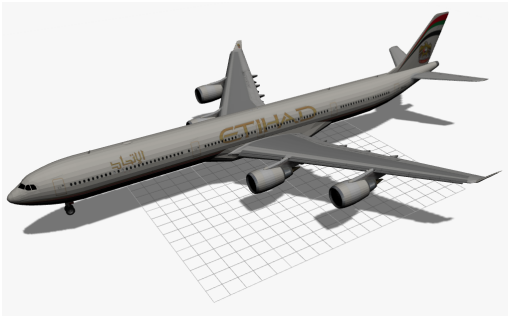


Figure 5: The aircraft used as the object of interest, chosen for its complex geometry

An aircraft was used as the object of interest, as shown in Figure 5 (Dimensions: $35 \times 30 \times 7$ m^3). The aircraft's complex geometry makes it an interesting case study; it has numerous curved surfaces, overhangs, occlusions and is large enough to demonstrate the scale of the problem being tackled. The aircraft also exhibits numerous symmetries that can be exploited.

The vehicle used is a UAV with two sensors, namely a laser scanner and RGB-D camera with specifications as in Table 1. The laser scanner covers a wide angle with a large range, making it suitable for obtaining many points during the large sweeping motions of the profiling stage. The laser scanner is mounted with a slight downwards pitch to ensure it can capture any skyward-facing surfaces. The RGB-D camera is used to obtain dense points at a closer range to create the highly detailed final reconstruction.

A circular collision box is constructed around the UAV to check for collisions with the structure, and its motion is constrained within a workspace of size $40 \times 40 \times 10$ m^3.

4.2 Procedure

Three main tests are performed; first, the classical NBV approach is performed without any profiling or predictive stage (Scenario 1). Second, profiling is performed, but no symmetry prediction is employed. The NBV approach is run using the profile as the initial map (Scenarios 2 & 3). Finally, both profiling and symmetry are performed before starting NBV (Scenarios 4 & 5).

Specification	Laser Scanner	RGB-D camera
Horizontal FOV	180°	60°
Vertical FOV	N/A (single layer)	45°
Resolution	0.25°	640×480 px
Range	0.1–30 m	0.5–8.0 m
Mounting orientation (Roll, Pitch, Yaw)	(0°, 10°, 0°)	(0°, 0°, 0°)

Table 1: Sensor specifications

To further enhance the coverage completeness, the CGVS and AGVS approaches are compared. In this context, CGVS means the value of α is fixed to zero in Equation 2. The value of α increments by one whenever the condition in Equation 6 is met for $N = 3$ and $\Delta E_{v,threshold} = 0.005$. Similarly, the termination condition is met if the Equation 6 is satisfied for $N = 5$ and $\Delta E_{v,threshold} = 0.001$.

For each scenario, the sensor begins in 5 predefined starting locations and the entire process runs until termination. The result of the 5 runs is averaged to give an indication of the average performance for each scenario.

5 Results

The final reconstructions are shown in Figure 6 and the results are plotted in Table 2, where the distance travelled, entropy reduction and coverage are compared. Coverage is evaluated by initializing two new occupancy grids: one constructed from the final point data and another from points sampled from the original mesh. The percentage of matching cells between the two grids is used to determine the coverage.

By changing the size of cells in the two grid, it is possible to measure two different types of coverage. Performing the matching with large voxels (eg. 0.5m) determines how much of the **overall shape** has been captured. On the other hand, matching smaller voxels (eg. 0.05m) gives an indication of how **detailed** the final model is.

Table 2 shows that compared to the full NBV approach (Scenario 1), the profile alone is able to capture the general shape of the structure (86.7% vs 86.9%) with lower travelling cost. However, since the profiling is performed from far away, the resolution is low and hence the density of the reconstructed structure is low (15.1% vs 73.3%).

When the NBV method uses the profile, we see that mid- and large-scale overage improve despite travelling roughly half the distance. While dense coverage did fall, it is mostly due to the lower number of iterations and distance travelled.

The dense coverage (resolution of $0.05m$) of the adaptive method consistently exceeds that of the constant grid method. This is mainly due to the vehicle being able to escape lo-

Scenario	Profiling	Symmetry Prediction	Viewpoint Sampling	Iterations	Distance (m)	Total Entropy Reduction	Coverage (Res=0.05m)	Coverage (Res=0.10m)	Coverage (Res=0.50m)
Profile	—	—	—	—	320.0	927,405	15.1%	61.6%	86.7%
1	No	No	CGVS	859	1,039.9	289,439	73.3%	78.7%	86.9%
2	Yes	No	CGVS	301	538.7	929,483	62.7%	89.0%	89.7%
3	Yes	No	AGVS	301	714.5	**929,509**	71.8%	88.0%	97.7%
4	Yes	Yes	CGVS	217	**526.6**	928,323	61.5%	87.7%	97.4%
5	Yes	Yes	AGVS	257	665.2	928,744	**86.5%**	**94.9%**	**99.5%**

Table 2: Results of the experiment

cal minima by being able to take larger steps. However, this caused travel distance to increase by approximately 30%.

Looking at entropy reduction, we can see that the profile alone is more effective at reducing entropy than the classical NBV method (Scenario 1). Since the LIDAR has long range and most of the space around the structure is empty, it converts most of the unknown cells to free cells. There are additional entropy reductions in Scenarios 2–5, but the value is much smaller as the methods are mainly observing occupied cells with a few scattered unknown cells.

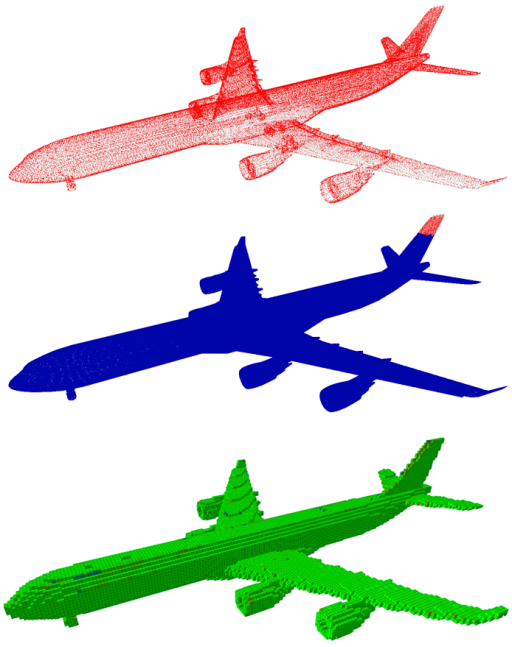


Figure 6: Final representation of the structure. **Top:** The profiled point cloud, representing the partial model obtained at the start of the process. **Center:** The final point cloud after NBV. Note the tail has not be visited during NBV¿ **Bottom:** The final occupancy grid, where green represents an occupied cell and blue represents unknown.

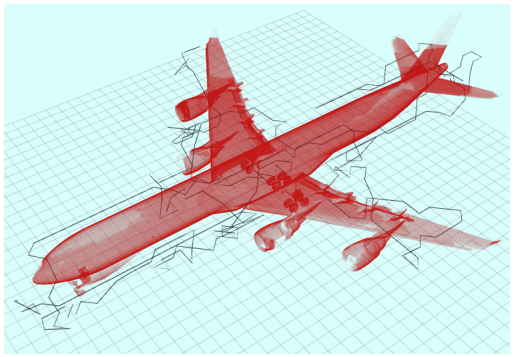


Figure 7: The trajectory taken by the drone during Scenario 4

6 Conclusions and Future Work

We have proposed a method of improving the Next Best View approach for the reconstruction of large, complex structures. A rough scan of the structure is performed to obtain its profile, which is then checked for mirror symmetry. The symmetry is used to make predictions about the shape of the structure to guide the NBV approach. We have also proposed a termination condition based on the average entropy change per cell, and an adaptive viewpoint sampling method based on the same principle.

The introduction of the profiling step shows improvement in coverage completeness with slightly reduced distance. The symmetry prediction performs similar to the classic method and occasionally outperforms it. Finally, the adaptive sampling method is able to escape local minima and brings the coverage completeness closer to 100% while given relatively dense reconstruction.

To further improve this method, we plan to develop a more holistic utility function by incorporating distance and/or density measures. Including the distance may help reduce the total distance travelled while targeting areas with low point density to improve the reconstruction density.

Another possible improvement is the inclusion of frontier methods. Be focusing directly on frontiers and creating viewpoints towards them, it may be possible to obtain a more

globally optimal solution.

REFERENCES

[1] Andreas Bircher, Mina Kamel, Kostas Alexis, Michael Burri, Philipp Oettershagen, Sammy Omari, Thomas Mantel, and Roland Siegwart. Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots. *Autonomous Robots*, pages 1–20, 2015.

[2] R. Almadhoun, T. Taha, L. Seneviratne, J. Dias, and G. Cai. Gpu accelerated coverage path planning optimized for accuracy in robotic inspection applications. In *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1–4, Oct 2016.

[3] Dirk Holz, Nicola Basilico, Francesco Amigoni, and Sven Behnke. Evaluating the efficiency of frontier-based exploration strategies. *Isr/Robotik 2010*, 2010.

[4] Simon Kriegel, Tim Bodenmüller, Michael Suppa, and Gerd Hirzinger. A surface-based next-best-view approach for automated 3d model completion of unknown objects. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4869–4874. IEEE, 2011.

[5] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova and R. Siegwart. Receding horizon "next-best-view" planner for 3d exploration. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, 2016. (accepted).

[6] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151. IEEE, 1997.

[7] Simon Kriegel, Christian Rink, Tim Bodenmüller, and Michael Suppa. Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects. *Journal of Real-Time Image Processing*, 10(4):611–631, 2015.

[8] Souhaiel Khalfaoui, Antoine Aigueperse, Yohan Fougerolle, Ralph Seulin, and David Fofi. Online fully automated three-dimensional surface reconstruction of unknown objects. In *The International Conference on Quality Control by Artificial Vision 2015*, pages 95340Z–95340Z. International Society for Optics and Photonics, 2015.

[9] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.

[10] Gavin Paul, Phillip Quin, Andrew Wing Keung To, and Dikai Liu. A sliding window approach to exploration for 3d map building using a biologically inspired bridge inspection robot. In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE International Conference on*, pages 1097–1102. IEEE, 2015.

[11] Lionel Heng, Alkis Gotovos, Andreas Krause, and Marc Pollefeys. Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1071–1078. IEEE, 2015.

[12] Phillip Quin, Gay Paul, Alen Alempijevic, Dikai Liu, and Gamini Dissanayake. Efficient neighbourhood-based information gain approach for exploration of complex 3d environments. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1343–1348. IEEE, 2013.

[13] Souichiro Oshima, Shingo Nagakura, Jeong Yongjin, Akihiro Kawamura, Yumi Iwashita, and Ryo Kurazume. Automatic planning of laser measurements for a large-scale environment using cps-slam system. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4437–4444. IEEE, 2015.

[14] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[15] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[16] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[17] Stefan Isler, Reza Sabzevari, Jeffrey Delmerico, and Davide Scaramuzza. An information gain formulation for active volumetric 3d reconstruction. 2016.

[18] Gavin Paul, Stephen Webb, Dikai Liu, and Gamini Dissanayake. Autonomous robot manipulator-based exploration and mapping system for bridge maintenance. *Robotics and Autonomous Systems*, 59(7):543–554, 2011.

[19] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

An Automated Rapid Mapping Solution Based on ORB SLAM 2 and Agisoft Photoscan API

Markus Bobbe*, Alexander Kern, Yogesh Khedar, Simon Batzdorfer and Ulf Bestmann
TU Braunschweig, Germany

ABSTRACT

This paper describes a system consisting of an UAV and a ground station capable of automated mapping based on aerial images. The focus of the presented system is to obtain georeferenced orthophotos within a short time frame.

Two approaches have been implemented in the system: an online visual SLAM based on ORB SLAM 2 and a photogrammetry pipeline using the Agisoft Photoscan API. Both approaches will be described and its result evaluated and compared.

1 INTRODUCTION

After natural disasters, responding forces rely on accurate maps to apply their ressources as fast and efficient as possible. Available maps and satellite images are often outdated or rendered useless due to the disaster (e.g. floods, earthquakes). UAVs can be used by first response forces to generate up to date georeferenced orthophotos. Aerial photogrammetry is a proven tool for the creation of such data. In recent years UAVs are used increasingly to generate the required images. Specialised software also simplified the photogrammetry workflow. Yet this workflow still requires knowledge regarding flight planning, data handling and processing parameter selection. The typical use case is the processing of survey data which is rarely time sensitive. The average processing therefore takes several hours since quality is often the most important criteria. Investigations regarding the best quality with minimal processing time are rare.

Another approach is the use of visual SLAM solutions which are capable of augmenting the solution incrementally with every new image. Available algorithms, often used in robotic reasearch, are less straightforward to use and have to be adapted for the specific use case. The results usually don't have the quality produced by photogrammetry solutions and investigations usually don't focus on the accuracy. Both approaches were adapted and implemented with the goal to gain a full automated rapid aerial mapping solution. This includes the mission planning, camera control, image transport to ground station, automated processing and the visualization of the results. The live mapping (SLAM) approach is based on the ORB-SLAM algorithm and updates the map when a new image is available. The photogrammetry based approach uses the commercial photogrammetry software Agisoft Photoscan. The processing is started with predifined parameters using the Photoscan API once the mapping mission is completed. In this paper the hardware and software setup for the developed system will be described and the generated results from both approaches will be compared.

In section 2 the hardware setup used for acquiring the images on-board the UAV and transferring them to the ground station is given. The implemented ROS based software infrastructure is also described. In section 3 the ORB SLAM 2 based solution is presented. The solution based on the Agisoft Photoscan API is given in section 4. Section 5 presents the mapping results generated by both solutions during flight tests and compares them regarding quality and speed.



Figure 1: Air Robot AR200.

2 SETUP

The setup is integrated in a modified Air Robot AR200 hexacopter displayed in figure 1. It carries a payload of 2.7kg leading to a maximum flight time of 25min. The payload consists of an Air Robot 2-axis camera gimbal (figure 2), an Intel Nuc i5 on-board PC, a experimental navigation package (consisting of 2 Analog Devices ADIS16488A IMUs, a uBlox M8T GNSS receiver and a Phytec Mira Cortex A9 processing board), a Gateworks GW 5520 wifi board and a AVT Manta 917G GigE Camera with a Cinegon 1.9/10mm lens. The ground station is equipped with the same wifi board and is based on an Intel i7 CPU and a GeForce GTX1080 GPU.

The systems uses ROS as the underlying framework. Figure 3 displays a flowchart of the implemented workflow. The

*contact: m.bobbe@tu-braunschweig.de



Figure 2: Gimbal with Camera and IMU.

sensor data is received by dedicated nodes on the aerial vehicle. The camera pose information is added to the image in the *Geo Image Flight Node*. The image is then sent to the *Geo Image Ground Node* on the Ground Station. The SLAM process is separated in the *SLAM Tracker Node*, which calculates the tranformation between images and the *Stitcher Node* which applies the transformations. The transformed images are displayed by the *Visualization Node*. The *Photogrammetry Node* receives the georeferenced images and triggers the photogrammetric processing ones the survey is finished. The results are also displayed in the *Visualization Node*.

3 SLAM Process

During the past few years computer vision especially SLAM based algorithms have undergone rapid development. Klein and Murray showed in their highly regarded work [1] how a novel pose estimation only by monocular image processing can look like and what great potential comes with it not only for augmented reality but the whole robotic scene. Integrating bundle adjustment and splitting up tracking and mapping into seperate threads were followed by a strong, realtime capable framework that had a significant impact on visual SLAM. However their approach lacked several, essential properties a modern SLAM needs for applications outside the academic use. Especially robust loop closure and relocalization was difficult to integrate into their framework, as they utilized separate features for tracking and place recognition. Mur-Artal and Montiel took up the basic principles proposed for PTAM and developed efficient solutions for these challenges. Their work led to the new designed ORB SLAM 2 [2] that reaches unprecedented accuracies and published it open source. It is briefly described in section 3.1.

By building up on that framework, we propose a pipeline to take advantage of the accurate camera pose estimated by the SLAM to generate large 2D aerial maps in realtime in section 3.2, similiar to those coming from modern photogrammetry software. A further comparison between these two approaches is then evaluated in section 5.

3.1 ORB SLAM 2 framework

The general structure of the ORB SLAM 2 framework is displayed in figure 4. Gray underlaid boxes represent seperate threads, while in the middle also main components of the implemented map and place recognition are shown.

Tracking

After initialization, which is explained in detail in [2], the tracking thread uses a constant velocity model to predict the current pose from the latest known position and movement. With this rough estimation a further analysis of features in the current region of interest is carried out. If the tracking state is triggered ”lost”, relocalization in the global map starts. After the tracking step a temporary pose is published and created keyframes are passed to the local mapping.

Local Mapping

In the local mapping thread a new identified keyframe is inserted as a node into a covisibility graph structure, which contains all relevant informations and relations as edges between the nodes. To achieve high accuracies after the tracking a local bundle adjustment is also carried out in this step.

Loop Closing

One of the main improvements of ORB SLAM 2, compared to PTAM introduced before, is the usage of only one type of features for all tasks within the framework. By taking advantage of a *bag of words approach* this allows the implementation of a place recognition to find loops in the graph and restructure it if neccessary. Based on that graph a global bundle adjustment is performed as soon as a loop is detected. In consequence a global consistent solution is achieved, even though the estimation error drifted over time.

3.2 Ortophoto Pipeline

After estimating the camera pose using ORB SLAM 2 framework in the next step these informations are used to gradually create a 2D orthophoto of the area during flight. To achieve this a lightweight image projection combined with georeferencing based on similiar transformation is proposed in this section.

3.2.1 Projection model

The projection model used in this paper follows the standard pinhole camera [3]. Therefore a world point $\underline{X} = (x, y, z)^T$ is described as

$$\underline{X} = s(\underline{\underline{RK}}^1 \underline{x}) + \underline{t} \tag{1}$$

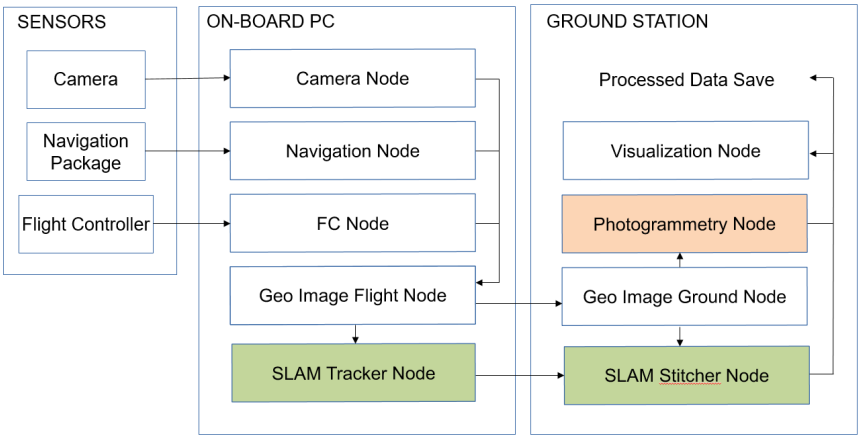


Figure 3: ROS node layout with SLAM (green) and Photogrammetry workflow (red).

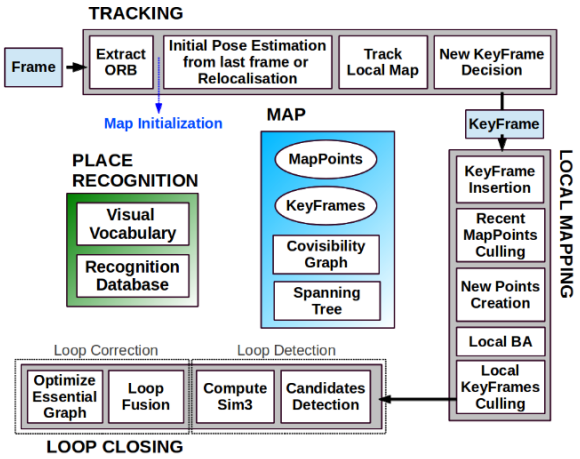


Figure 4: ORB SLAM 2 framework structure [2].

where $\underline{\underline{R}}$ is the 3x3 camera rotation in the world frame, $\underline{\underline{K}}$ the 3x3 camera intrinsic calibration matrix, s the scale factor of the projection direction vector, $\underline{x} = (u, v, 1)^T$ a point in the image plane in homographic coordinates and $\underline{t} = (t_x, t_y, t_z)$ the exterior position of the camera. The model is further extended by additional parameters k_1, k_2, p_1, p_2 and k_3 to compensate geometric distortion caused by the lens. Usually image points are therefore undistorted first by following the equations

$$x_{corrected} = x + [2p_1xy + p_2(r^2 + 2x^2)] \quad (2)$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2xy] \quad (3)$$

and projected applying the proposed formulation in 1 afterwards.

3.2.2 Image Projection

To create a 2D orthophoto from 3D camera poses a definition of the actual common reference plane is needed. The reference plane must be orthogonal to all camera observations to ideally reduce perspective errors to zero. Even when using gimbal stabilized data and a low distortion lens this error can not be completely avoided. Therefore an orthogonal reference plane can only be approximated. In this implementation this problem was solved by collecting a minimum amount of measurements in a buffer and calculating the best fitting solution of the sparse pointcloud generated by ORB SLAM 2. This approach is only valid as long as the ground is flat and the flight altitude relatively high. Applying additional RANSAC formulation reduces further influence of outliers. As soon as a valid reference frame is estimated for the first measurements, it is kept fixed for the rest of the series. Therefore large misalignements in the beginning can not be compensated afterwards. In the future it might be beneficial to check validity of the reference plane periodically and recalculate the current map. In the next step image boundaries can be projected into the reference plane by using the formulation of section 3.2.1. With these four points an exact perspective transformation with 8 DOF can be calculated. It is then further applied on the whole image and stitched to the current global map.

3.2.3 Geo-referencing

Simultaneously with the calculation of the common reference plane an additional geo-referencing for the images is estimated. It allows to transform every pixel of the final map to a defined lat/lon-coordinate. To achieve this, all images are geotagged in the moment they are captured with the latest available GNSS informations. After choosing the first complete measurement consisting of visual pose and global

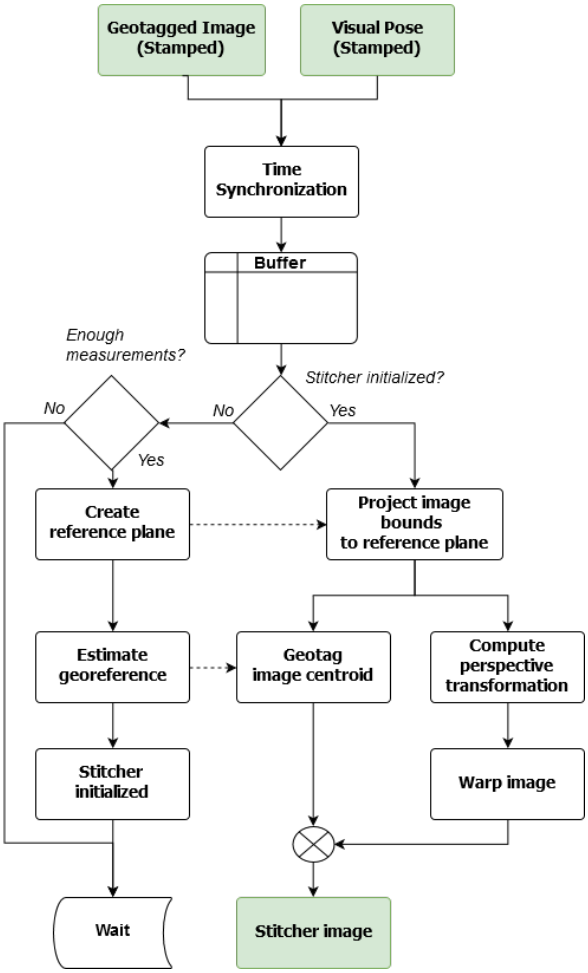


Figure 5: Structure of the orthophoto stitching.

GNSS position each as root of a local coordinate system, the following steps are performed for every image in the initial buffer:

1) Calculate image centroid in common reference plane

2) Transform centroid to local coordinate system (relative to centroid of first measurement)

3) Convert GNSS position from WGS84 to UTM32

4) Transform GNSS position to local coordinate system (relative to GNSS position of first measurement)]

Afterwards a 4 DOF similiar transformation (x, y, rotation, unitary scale) between these two local coordinate systems can be calculated by linear programming. Considering the root of the local coordinate systems a transformation to the global world is also possible.
While this approach is straighforward, it induces several uncertainties. By projecting the centroids in the reference plane and assuming their plane position matches the global position measured in the air, an error linked to the stabilization quality of the gimbal can be expected. Depending on the update rate of the GNSS receiver and the FPS of the camera different images can be tagged with the same position leading to a bad initialization. Therefore only the first appearence of every global position is considered in the described process. However asynchronous time between geotagging and image capturing results in an additional error depending on the speed of the UAV. At last by converting WGS84- to UTM32-coordinates affine transformation during linear programming can be avoided. This allows more intuitive handling of the coordinates, but also applies an uncertainty of several centimeters depending on the size of the map.

4 Photogrammetry Process

The photogrammetry process was implemented using Agisoft Photoscan 1.2.6. Extensive studies regarding the accuracy attainable by the software have been done before, for instance by Gini et al. [4] and Ouedraogo et al. [5]. Yet studies which focus on the minimal processing time are not available. In this implementation the Photogrammetry Node saves images belonging to the current mission and loads them into Agisoft Photoscan using the API once the survey mission is finished. The camera location and the inner camera geometry were written to the EXIF file of each image by the *Geo Image Ground Node* to enable its usage to georeferencing the results. The process is started using preselected processing parameters. The parameters affect the required processing time and the quality of the result.

The process consists of 4 steps. During the camera alignment features are identified and compared to optimize the homographic equation and determine the camera locations. In the next step a mesh is generated using the generated tie points. The mesh is then used to create an orthophoto and to export it in the desired format in the last step. It is possible to use a dense cloud or a Digital Elevation Model as the basis for the orthophoto generation. Yet the creation of these models would significantly increase processing time and was therefore rejected.

To determine a reasonable compromise between quality and processing time, 4 profiles with different quality parameters were created and and tested in the next section. The relevant parameters of each profile are given in table 1.

profile	alignment accuracy	mesh face count
Agi lowest	lowest	low
Agi low	low	low
Agi medium	medium	medium
Agi high	high	high

Table 1: Photogrammetry parameter profile defentition.

5 Evaluation

To demonstrate the implemented workflow and to validate and compare the created results the complete system was tested by flying a sample mission. To estimate the accuracy of the created maps, 8 ground reference points (GRPs) were distributed over the mission area. The location of the GRPs were determined using a RTK-GNSS system leading to a horizontal RMSE below 2cm. To enable robust processing the overlap and the sidelap was chosen to be 70%. This resulted in a creeping line mission consisted of 4 times 130m lines with a distance of 25m and an altitude of 100m over ground.

The image rate and therefore the frontlap varies between the two implementations. The SLAM node receives the images with a higher framerate which is benefical for the tracker. The processing power of the earlier described ground station enables processing with a frame rate of up to 4Hz. This leads to an frontlap ov 99% and 864 images, yet only 27 were used as keyframes. The photogrammetry node receives the images with a lower framerate to limit the total number of images to minimize processing time. The chosen framerate of 0.3Hz leads to an overlap of 85%. In total the photogrammetry node received 47 images in less then 3 minutes flight time.

The photogrammetry pipeline was successfully tested in the field. To compare the presented profiles, they were triggered one after another. The created results are displayed in figure 8. All profiles provided consistent solutions and were succesfully georeferenced. The visual comparison gives no significant differences. The remaining profiles let to similar results also without notable differences to visual inspection. The processing time varied between 0.9 and 3.2 minutes. A comparison of this and other criteria is given in figure 6. The calculated position of the GRP was determined in each orthophoto. The derivation to the reference measurement is given in table 2. While the mean error in the low profile is two thirds of the error in the lowest profile, the calculated errors using the medium and high profiles are not enhancing significantly. The number of tie points created by the lowest profile is an order of magnitude lower compared to the other three profiles. The reprojection error roughly halves with each profile step. The xy camera error describes the mean horizontal camera displacement regarding the inital GNSS location and the final calculated position and is almost identical in all profiles.

To accelerate the photogrammetry process, the resolution of the created orthophoto can be reduced. Figure 9 displays the total processing time using the lowest profile with different target ground sampling distances (GSD) during orthophoto creation. The processing time can be reduced significantly and reaches processing times of 7 seconds when a orthophoto with an GSD of 0.4m is created.

The created final image of the SLAM pipeline is displayed in figure 8. The image was updated with every new image and was therefor finished before the copter landed. It reaches a GSD of 0.05 m, while the mean location error mea-

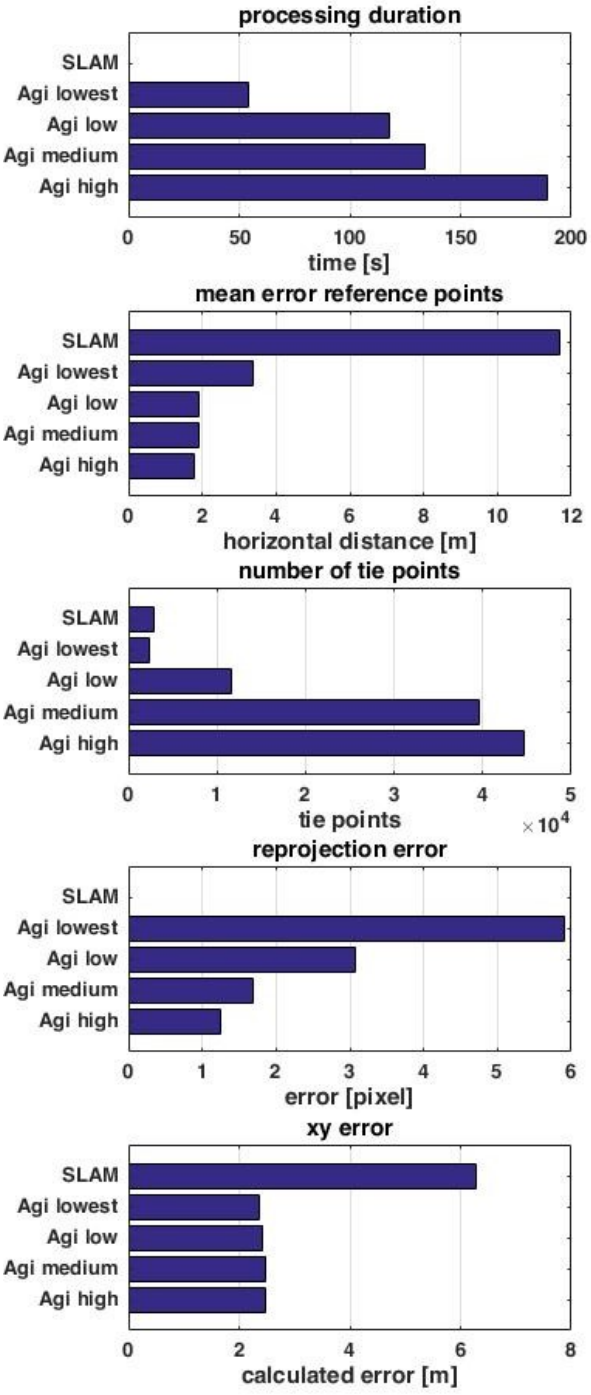


Figure 6: Mapping evaluation results.

GRP	1	2	3	4	5	6	7	8	mean
SLAM	15.83	18.94	14.56	16.57	9.21	10.15	4.65	4.73	11.67
Agi lowest	2.49	2.19	3.40	3.17	4.03	3.80	4.58	3.66	3.39
Agi low	1.41	1.28	1.87	1.74	2.19	2.06	2.49	2.46	1.91
Agi medium	1.34	1.13	1.80	1.70	2.15	2.13	2.64	2.63	1.91
Agi high	1.17	1.08	1.64	1.65	1.94	1.99	2.38	2.41	1.77

Table 2: Vertical error at reference points in meter.



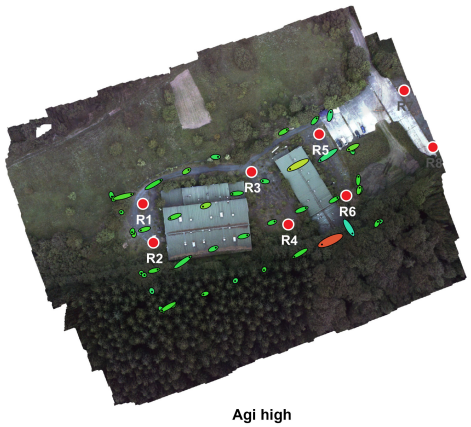


Figure 7: Orthophotos created with the profiles lowest and high



Figure 8: Orthophotos created with SLAM approach

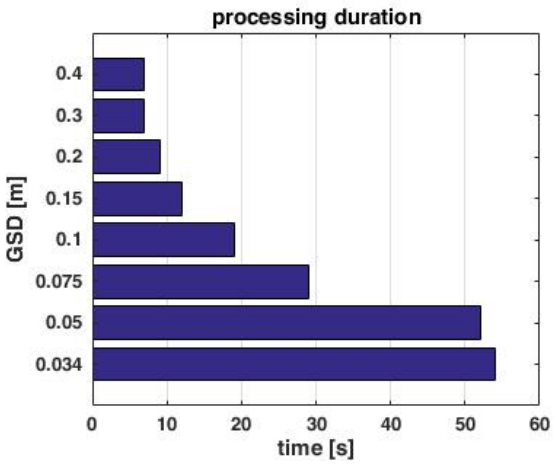


Figure 9: Processing Times using the Agi lowest profile and different GSD.

sured using the reference points was about 12m which is significantly larger than the errors observed in the photogrammetry results. This and other results are also given in figure 6.

6 Conclusion

The introduced system demonstrated the proposed capabilities regarding automated camera operation, image transport and ortophoto generation. Both implemented mapping pipelines generated adequate results which fullfilled the requirements in a disaster scenario. The SLAM approach is capable of delivering instant results but the overall georeferencing accuracy is roughly one order of magnitude lower compared to the photogrammetry pipeline. The comparison of the different photogrammetry profiles indicated that the usage of the second fastest profile low is recommended for most scenarios since it is the fastest profile that is capable of reconstructing the complete area. Also the more elaborate profiles do not lead to significantly better results regarding georeferencing accuracy.

Acknowledgements

The presented work was done within the joint research project ANKommEn funded by the German Federal Ministry of Economic Affairs and Energy administrated by the Space Administration of the DLR (funding code: 50NA1518).



References

[1] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[2] Mur-Artal Raúl Montiel, J. M. M. and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[4] Daniele Passonib Livio Pintob Giovanna Sonaa Paolo Dossoc Rossana Ginia, Diana Pagliarib. UAV Photogrammetry: Block triangulation comparison. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1(W2), 2013.

[5] Charles Debouche Jonathan Lisein Mohamar Moussa Oudraogo, Aurore Degr. The evaluation of unmanned aerial system-based photogrammetry and terrestrial laser scanning to generate dems of agricultural watersheds. *Geomorphology*, 214:339–355, 2014.

Investigation on Boundary Layer Ingestion Propulsion for UAVs

L. Teperin,* M. El-Salamony,† A. Moharam,‡ and M. Shehata,§
* Central Aerohydrodynamic Institute (TsAGI),
Zhukovsky street, 1, Zhukovsky, 140180, Russia.
Moscow Institute of Physics and Technology,
Department of Aeromechanics and Flight Engineering,
Gagarina street, 16, Zhukovsky, 140180, Russia.

ABSTRACT

Power reduction is one of the most current important issues. One promising way is to use boundary layer ingestion propulsion. In order to evaluate the benefits of using boundary layer ingestion propulsion in reduction of power consumption and to understand the difference between the propeller placed behind and in front of a fuselage, a comparison is made between propeller placed behind and in front of a fuselage. It is found that the backward position has less drag and power consumption. Power saving coefficient reached 21.7% compared to the forward position and drag coefficient is 28.5% less. Also it shows tendency to stabilize and prevent separation of the boundary layer.

1 INTRODUCTION

Current trends in unmanned aerial vehicles (UAV) market heads to power reduction. One promising method is boundary layer ingestion (BLI) propulsion. The fundamental principle of BLI propulsion is that a propulsor ingests and reaccelerates airframe boundary layer which reduces the wake deformation downstream and jet kinetic energy for the same net force which decreases the power to be added to the flow by propulsor.

The interaction between a propeller and a body has both benefits and drawbacks. The advantages on power consumption can be explained using Froude model of propeller. For the same thrust, the propeller consumes less power if the incoming flow velocity is less. By placing the propeller in a boundary layer or a wake of a body, the incoming flow will have less velocity compared to the free stream, so the power consumption decreases and propulsive efficiency increases. Efficiency can even exceed 100% since the reference velocity is the free stream velocity which is higher than the actual incoming velocity to the propeller. Another useful point is that the power is a function of the difference between inlet and outlet velocities squared, this counted as a benefit for the power reduction since the incoming flow is not uniform, so the exit velocity can be shaped to give the minimum power consumption.

On the other hand, imposing a propeller in the vicinity of a lifting body will affect the boundary layer and the pressure distribution. Air suction near the trailing edge will create low pressure zone and so the pressure drag on the body will increase and the flow will accelerate. Flow acceleration will increase the shear force and so the friction drag will increase. Also the angle of attack must increase to compensate the losses of lift, which also will increases drag. Losses in lift occurs due to the acceleration if the flow under the body, which will decrease the pressure difference in the vertical direction and consequently lift decreases. These effects hold for axisymmetric bodies except the fact that it generates no lift.

Regarding the effects on the boundary layer and the laminar-turbulent transition, this region will be shifted downstream due to presence of negative pressure gradient zone near the trailing edge, and hence, drag will decrease, but this effect is negligible compared the increment of friction drag due to flow acceleration. By using some numerical investigations using panel method combined with integral boundary layer equations, it was found that BLI propulsor under some considerations can prevent boundary layer separation.

This article considers the integration between fuselage and BLI propeller and their mutual effects on the power consumption, drag, and boundary layer properties for the case of cruise flight. It answers two main questions: is it advantageous to integrate them from the power consumption point of view and what is the impact of the propeller position on the flow field.

To understand the effect of the BLI propeller on the power saving, two propeller positions are studied: behind and in front of the fuselage. To study the aerodynamic effect of the BLI propeller on the body, a comparison is conducted between body with and without BLI propeller.

2 LITERATURE REVIEW

Integration between propulsor and a body was studied theoretically in many papers. Smith [1] proved analytically that for the case of a flat plate, propulsive efficiency working

*Email address: teperin@mail.ru
†Email address: elsalamony.mostafa@phystech.edu
‡Email address: a.moharam@phystech.edu
§Email address: moamen.shehata@phystech.edu

on the wake of a flat plate is 127% and power saving about 20%. Teperin and Ujuhu of TsAGI [2] investigated the effect of interference with propeller glider aircraft without considering viscosity. Later, a similar problem has been solved numerically for a propeller mounted at the stern of an airship [3]. In [4] the useful interference between free stream propeller and wing is considered in more detail. Recently, Teperin et al. [5] considered the optimal pressure difference distribution across the propeller to decrease power consumption, and the results shown that the power saving reached more than 20% in comparison of propeller with uniform pressure difference. Drela of MIT [6] derived an analysis of compressible viscous flow around a body with engine based on the mechanical power and kinetic energy instead of the regular forces and momentum flow method, and he explained how to quantify the boundary layer and wake ingestion benefits.

Many experiments were carried out on several bodies as airships where Stern-mounted propellers were investigated as a way to improve the propeller efficiency significantly and reduce its power consumption, but they are directed towards airships and large aircraft. In the experiment of McLemore of NASA [7] they achieved propulsive efficiency of 103%. Goldschmit of NASA [8] found that 50% of power could be saved using BLI and counter rotating propellers as in the experimental aircraft Douglas XB-42 Mixmaster.

Concerning the current development for the future aircraft, the conceptual project of Double Bubble (D8) of MIT, NASA, and Boeing [9] has fuel saving of 33% compared to the optimized conventional configuration having the same technology level using BLI and another techniques. ONERA [10] also achieved 23% of power saving in wind tunnel experiments for a similar project with Airbus.

Interaction between BLI propulsor and airfoil is considered in [11]. Main findings are that BLI affects mainly the pressure-dependent parameters. It is found that drag increased for the given flight conditions 11.4% divided as increment in friction drag 2.11% and 16.07% in pressure drag and lift decreased 8.89%. On the other hand, the power is reduced by 14.4% as compared to the free-stream propulsor, and the propulsive efficiency reached 109% and it was concluded that BLI is useful from the viewpoint of power saving

Similar work is done in [12] in more detail. it was found that boundary layer became more laminar and the boundary layer thickness decreased, the momentum and energy losses decreased, the flow in the leading part of the airfoil is not changed, and the friction drag can be neglected with respect to pressure drag.

The optimal distribution of the pressure difference across the active disk of propeller for minimum power consumption is developed in[5]. It was depicted that propeller with optimum pressure difference placed in the rear part of airship shows reduction in consumed power more than 20% when compared to propeller with uniform pressure difference.

Numerical investigations made by Lutz T. et al. [7] using panel method combined with integral boundary layer equations found that BLI propulsor under some considerations can prevent boundary layer separation.

Considering experiments for small bodies, an experiment was conducted by Sabo and Drela in MIT [13] as a proof of concept for the boundary layer ingestion concept. Electric ducted fan propulsor behind a NACA 0040 body of revolution is examined at a Reynolds number of approximately 2.4e5. Wind tunnel air velocity was 13.4 m/s and length of the body was 0.25 m. Measurements showed power savings reached 26% for untripped flow case and 29% for tripped flow case. Also it is found that power saving increases when the propeller is placed closer to the body and when the propulsor is aligned in the same axis of symmetry for the body.

Another experimental study conducted in Lv et. al in TU Delft [14] using the Stereoscopic Particle Image Velocimetry for the first time to visualize the flow field at the location of interaction between a propeller and an incoming body wake and to quantify the terms of power balance method. Results show that one of the main mechanisms responsible for the claimed efficiency enhancement in the experimental setup is due to the utilization of body-wake energy by the wake-ingesting propeller. Shaft power was reduced in the case of wake ingestion by 10% and by 18% for Boundary layer ingestion.

3 Theoretical Approach

3.1 Free stream propeller

Froude model can be used for modelling the free stream 2-D propeller – shown in Figure 1– where the propeller is a zero-thickness disk that creates pressure difference converted at the far field to velocity difference producing thrust. The flow is assumed to be axisymmetric, inviscid, without mixing at the jet edges. Thrust is calculated by the following integration[1]:

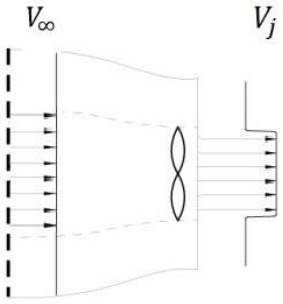


Figure 1: Constant inlet conditions for propeller

$$T = \iint \rho * V_j * (V_j - V_\infty) dA \tag{1}$$

For constant parameters this integration leads to

[1] It is notable that all of the integrations in this section are made on Trefftz plane where pressure is equal to the undisturbed upstream pressure.

$$T = \dot{m} * (V_j - V_\infty) \tag{2}$$

Propulsive power added is the difference between the kinetic energies of the stream tube passes by the propeller downstream and upstream the propeller, it is calculated as

$$P_p = \iint \frac{1}{2} * \rho * V_j * (V_j^2 - V_\infty^2) dA \tag{3}$$

And for constant parameters this integration leads to

$$P_p = \frac{\dot{m}}{2} * (V_j^2 - V_\infty^2) \tag{4}$$

The propulsive total power (Equation 3) can be decomposed by the Power Balance Method [6] into two categories; thrust (useful) power which is represented by multiplication of thrust by the incoming velocity Equation (5) and wake power due to the velocity perturbations downstream. This wake requires excess energy to be flattened by the flow viscosity. This power is directly proportional to the velocity difference as shown in Equation 6.

$$T * V_\infty = \iint \rho * V_j * V_\infty * (V_j - V_\infty) dA \tag{5}$$

$$E_{wake,prop} = \iint \frac{1}{2} * \rho * V_j * (V_j - V_\infty)^2 dA \tag{6}$$

Propulsive efficiency is defined as the useful power thrust multiply velocity divided by propulsive power (total power). By some mathematics we get the well-known Froude formula

$$\eta = \frac{2 * V_\infty}{V_j + V_\infty} \tag{7}$$

And considering the power decomposition we get

$$\eta = \frac{T * V_\infty}{T * V_\infty + E_{wake,prop}} \tag{8}$$

It is notable that the actual power is more than the calculated power because of the viscous and induced losses.

3.2 Drag Characteristics

Drag is the momentum losses between upstream and downstream flows of a stream tube passes around a body according to the momentum equation [15] as shown in Figure 2. Therefore, it can be calculated directly from the velocity distribution of the wake by the following formula

$$D = \iint \rho * V_w * (V_w - V_\infty) dA \tag{9}$$

This force can be represented in form of power consumed as drag multiplied by the upstream velocity

$$D * V_\infty = \iint \rho * V_w * V_\infty * (V_w - V_\infty) dA \tag{10}$$

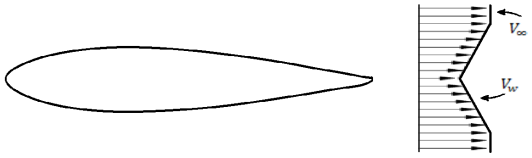


Figure 2: Airfoil and velocity distribution of the wake.

This power is consumed due to two reasons: dissipation in the viscous boundary layer and velocity perturbation in the wake. The dissipated energy in the boundary layer is quantified as the energy losses between upstream and downstream in the stream tube that passes around the airfoil, which is equal to:

$$\phi_{BL} = \iint \frac{1}{2} * \rho * V_w * (V_\infty^2 - V_w^2)) dA \tag{11}$$

While the energy of the wake can be quantified by the perturbations in the stream tube and calculated by the following equation:

$$E_{wake,body} = \iint \frac{1}{2} * \rho * V_w * (V_\infty - V_w)^2 dA \tag{12}$$

And hence, the total consumption in kinetic energy is the summation of the BL dissipation and the wake perturbations which are equal to the drag power

$$D * V_\infty = E_{wake,prop} + \phi_{BL} \tag{13}$$

3.3 Ideal BLI Propulsor

The concept of the ideal Boundary Layer Ingestion Propulsor is to make use of the BL air and reenergize it to generate the same amount of thrust but by less power. In addition, the BLI propulsor will 'fill' the momentum gap generated by the body due to drag and so it will minimize the wakes created by the body and the propeller downstream which will decrease the losses in kinetic energy very much as shown in Figure 3. Moreover, the free stream propulsor obtain thrust by accelerating the free stream which will generate propulsor wake downstream which absorbs energy also.

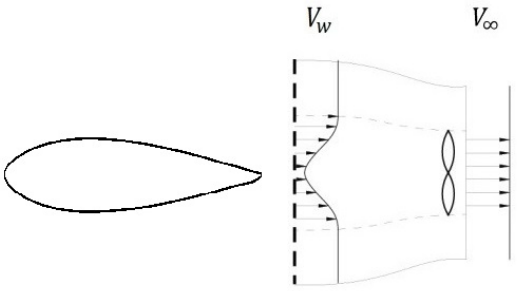


Figure 3: Perfect boundary layer Ingestion.

In the ideal case, all the wake will be ingested and flattened, the jet velocity will exactly match the undisturbed velocity. Thus, the only energy dissipated will be due to the BL viscosity which is defined by Equation 11. Since the stream tube is accelerated to the upstream conditions and all the wake is eliminated, the kinetic energy added by the propulsor to the stream tube will be equal only to the losses in the boundary layer, which can be evaluated as the difference in kinetic energy of the flow in front of and behind the body as in the following equation:

$$E_{prop,BLI} = \iint \frac{1}{2} * \rho * V_w * (V_\infty^2 - V_w^2) dA \qquad (14)$$

So the required power decreased while the drag and thrust still exist with the same value.
One main important parameter used to quantify the benefits of BLI is the power saving coefficient which is defined as in the following equation

$$PSC = \frac{P_{noBLI} - P_{BLI}}{P_{BLI}} \qquad (15)$$

Concerning the efficiency of this system, the propulsive efficiency is still defined as the useful power (thrust multiply velocity) divided by total power. Since thrust is equal to drag, we can replace the numerator to be $D * V_\infty$. By using Equation 13 the efficiency in will be

$$\eta = \frac{D * V_\infty}{E_{prop,BLI}} = \frac{\phi_{BL} + E_{wake,prop}}{\phi_{BL}} > 1 \qquad (16)$$

4 Problem Description

Investigation is conducted using Computational Fluid Dynamics package Ansys CFX 17.2 using RANS equations and a Shear Stress Transport (SST) turbulence model.

4.1 Geometry

The fuselage is modelled as axisymmetric body of NACA 0024 cross section with length of 0.5 m and the propeller is modelled using active disk theory with radius is 0.05 m, which corresponds to 10% of length, placed with two configurations: 0.02 m behind the trailing edge of the fuselage, and 0.02 m in front of the fuselage which corresponds to 4% of chord.

4.2 Mesh Description

A cylindrical domain was considered of width 6 m and height of 2 m revolved by 1°. The fuselage is placed 2 m apart from the domain inlet. A body of influence is made with dimensions of 1.9 m width and 0.52 m height and the distance between the fuselage nose and the rectangle is 0.52 m as shown in Figures 4 and 5.

The airfoil has number of divisions equals to 200 and the propeller has 40 for each edge. Element size in the body of influence is 0.005 and the growth rate is 1.04. The mesh has

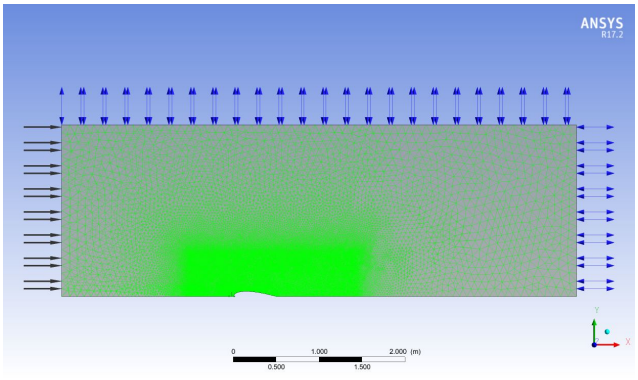


Figure 4: Computational domain for the studied case.

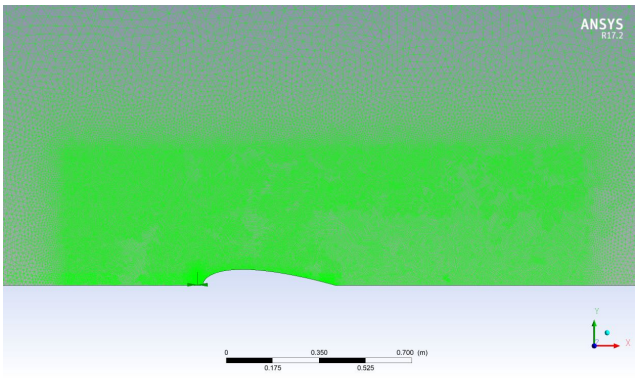


Figure 5: Body of influence around the studied body.

total number of nodes and elements 149401 and 521680 respectively. Shown in Figures 6 and 7 the mesh around the body and the active section in forward and backward configurations respectively.

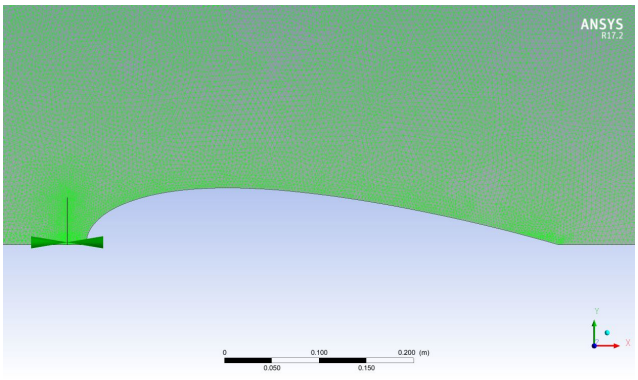


Figure 6: Computational mesh around the body and the active section in forward configuration.

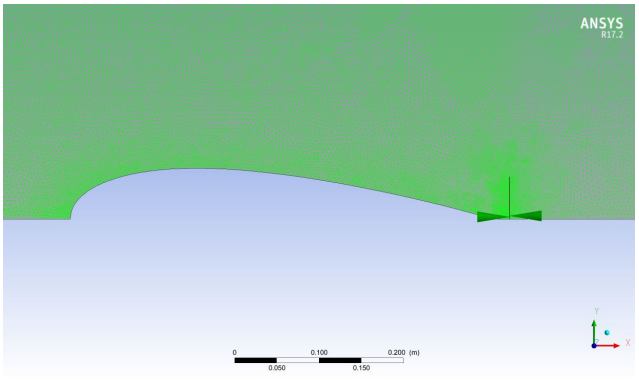


Figure 7: Computational mesh around the body and the active section in backward configuration.

4.3 Boundary Conditions

CFD calculations are sensitive to the boundary conditions. These conditions affect the stability and convergence of the solution and have impact on the physics of the simulated case. The BCs must generate flow compatible with the physics of the flow and the theory used for modelling.
The boundary condition of the body is no-slip wall. Propeller is set to have pressure difference boundary condition to provide thrust equals to drag. Pressure difference across the propeller is calculated by Equation 17.

$$\Delta P_i = \Delta P_{i-1} + \frac{D-T}{S_{prop}} \times K \tag{17}$$

where ΔP_i and ΔP_{i-1} are the pressure difference across the propeller in the current and previous iterations, S_{prop} is the propeller area and K is a relaxation factor. All boundary conditions are listed in the table below.

Boundary	Condition
Inlet	Velocity Inlet
Side Wall	Symmetry
Fuselage	No-Slip wall
Propeller	Pressure difference
Exit	Opening

Table 1: Boundary conditions

Pressure and temperature are set to be 1 bar and 15°C respectively. Free stream velocity is 10 m/s (to be far from the compressibility effect). Angle of attack is kept zero since only the axisymmetric case is considered.

5 Results and Discussion

For the forward propeller position, thrust is set to be equal to the profile drag. Equation 3 can be used to find the required power. The results show that required power is 1.008 W, mass flow rate= 0.0828 kg/s, pressure ratio = 1.25 and the drag coefficient is 0.0147 divided to 57.12% of pressure drag and 42.88% of friction drag.
Same procedure is also done for the backward propeller position. The results show that required power is 0.792 W, mass flow rate= 0.091 kg/s, pressure ratio = 1.18 and the drag coefficient is 0.0104 divided to 50.5% of pressure drag and 49.5% of friction drag.
By comparing the two configurations, it is found that the propeller located after the fuselage will cause less drag (28.5% less). The consumed power is also decreased and the power saving coefficient reached 21.74%.Moreover, BLI ingests more mass flow rate and apply less pressure ratio which is beneficial for the durability of the propeller.

Position	C_D	Energy $[W]$
Backward	0.0104	0.799
Forward	0.0145	1.021

Table 2: Results for backward and forward positions.

For better understanding of the drag increment reason, it is important to investigate how the flow will go past the body. For the forward position, the flow acceleration in the front part of the body is high and reached its maximum value in the maximum thickness position, then the stream tube diverge as its area increases downstream. Due to this a positive pressure gradient is established which eventually leads to flow separation after the maximum thickness position and flow unsteadiness as shown in Figure 8 which leads to massive increment in drag.

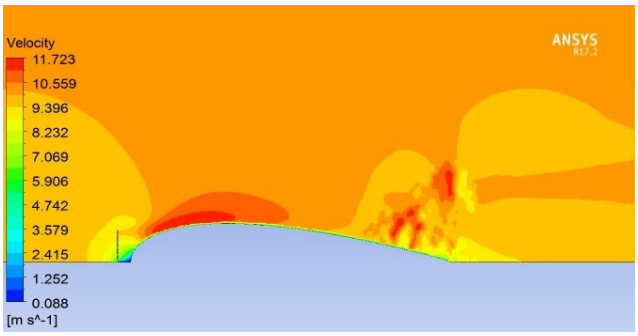


Figure 8: Velocity contour around the body with forward propeller showing unsteady flow.

On the other hand, the backward propeller position, the flow is not forced to accelerate in the front part of the body, it will accelerate in the rear part due to presence of negative pressure gradient created by the propeller. This leads to flow attachment in the rear part of the body as shown in Figure 9.

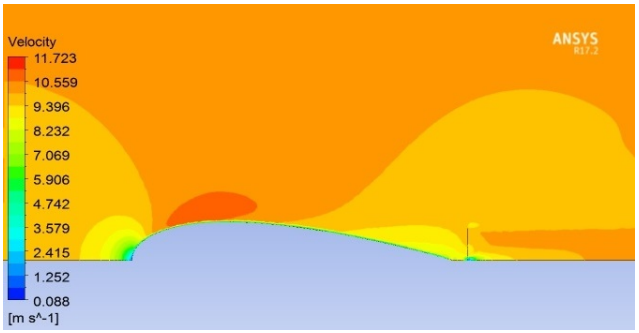


Figure 9: Velocity contour around the body with backward propeller.

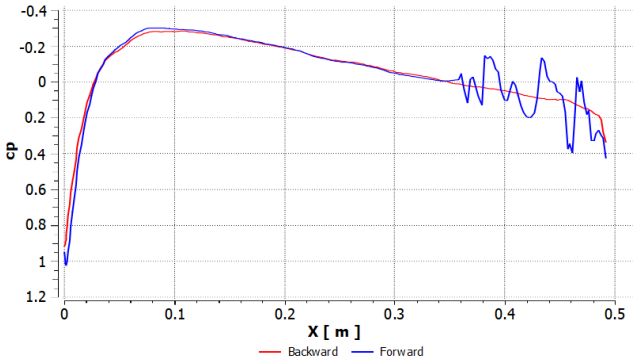


Figure 10: Pressure coefficient distribution on the body wall.

Energy consumption is highly dependent on drag and incoming velocity. For the backward configuration, the required power is less compared to the forward configuration. This is due to two reasons: the backward configuration requires less thrust due to absence of flow separation and the incoming velocity is less as it is affected by the boundary layer of the upstream body. In other words, the stream tube ejected from the propeller does not affect the upstream body in contrast to the forward configuration which will stimulate the flow separation after the maximum thickness position of the body.
Of interest the pressure distribution on the body surface is investigated. As shown in Figure 10 for the case of the forward configuration, the leading edge for the forward configuration faces higher pressure than the backward pressure because of the propeller jet is concentrated in this zone as shown in Figure 11. Near the maximum thickness position the forward configuration has higher speed and hence less pressure is applied. Further downstream there is massive pressure disturbance in the rear part of the body as in illustrated in Figures 12 and 13 where velocity contours around the trailing edge of the body are shown for both configurations. Contrarily the backward configuration stabilizes the flow and accelerates the boundary layer and decreases its thickness as shown in Figure 14.

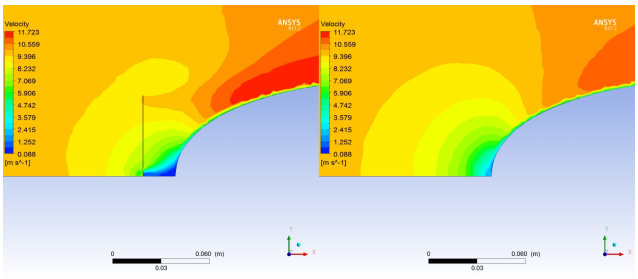


Figure 11: Velocity contour around the nose with forward and backward propeller.

Distribution of velocity components in the trailing edge show big scattering for the forward configuration while the backward one shows smooth behaviour as shown in Figures 15 and 16.

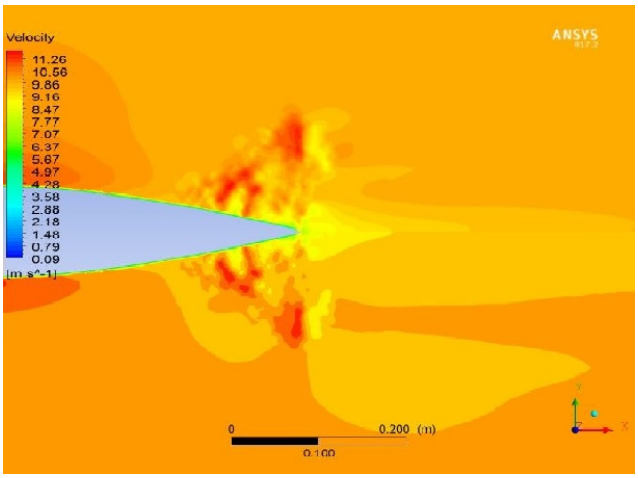


Figure 12: Velocity contour around the tail with forward propeller.

Of interest to study the Power balance Method terms for the backward configuration. Considering the drag terms, BL dissipation ϕ_{BL} was measured according to Equation 11 and it is equal to 3.9226 W. Wake energy $E_{wake,body}$ is evaluated according to Equation 12 and it is found that it equals to 0.102 W. Drag power according to Equation 10 is 4.0246 W which is equal to adding ϕ_{BL} and $E_{wake,body}$ as state in Equation 13. Power consumed by the BLI propeller $E_{prop,BLI}$ is

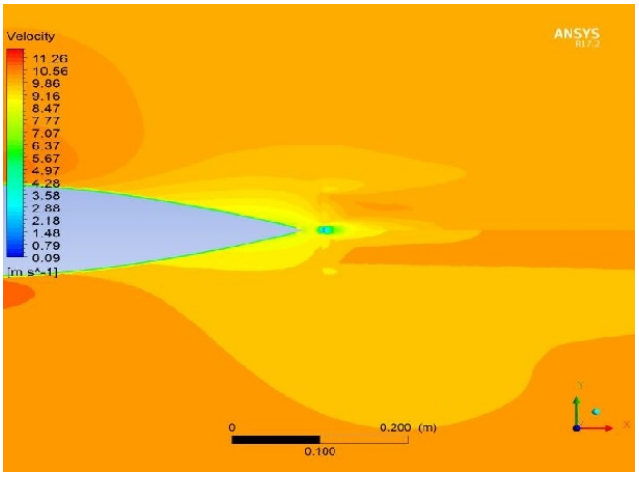


Figure 13: Velocity contour around the tail with backward propeller.

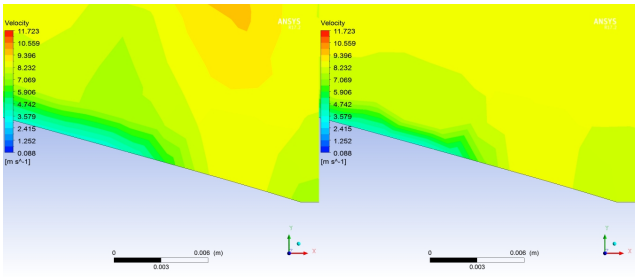


Figure 14: Velocity contour around the tail with forward and backward propeller showing the smaller boundary layer for the backward configuration.

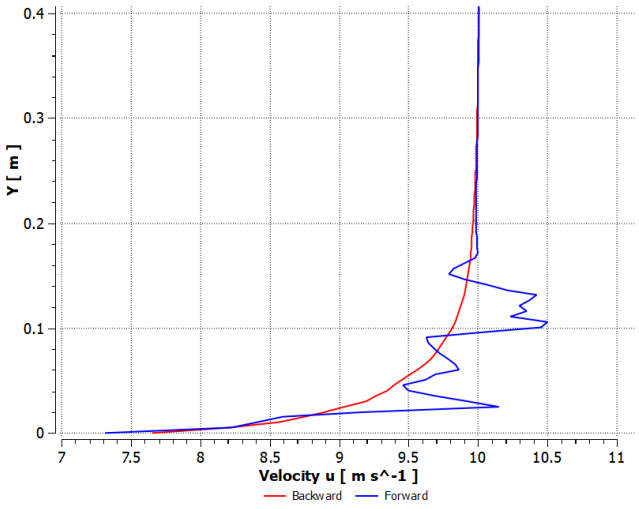


Figure 15: Horizontal velocity component distribution behind the trailing edge.

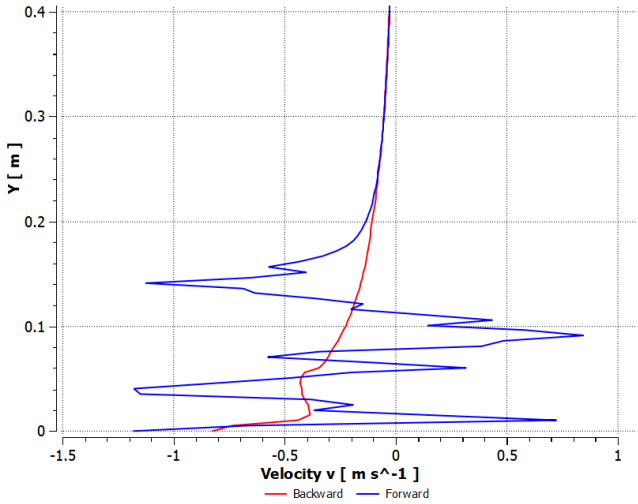


Figure 16: Horizontal velocity component distribution behind the trailing edge.

calculated as in Equation 14, and it equals 3.9804 W. the error between $E_{prop,BLI}$ and ϕ_{BL} is due to a problem in mapping the velocities to Trefftz plan because there is a small zone near the center line of the propeller has negative pressure, which led to imaginary values. it is notable to mention that the integration is made in a line not in a circle. Efficiency (using Equation 16) is calculated and it is 102.6%.

6 Conclusion

Benefits of the boundary layer ingestion and its impact on the aerodynamics of an axisymmetric body are investigated. A comparison is made between propeller placed behind and in front of a fuselage. It is found that the backward position has less drag and power consumption. Power saving coefficient reached 21.7% compared to the forward position. The forward propeller has higher drag because it increases the disturbance in the flow after the maximum thickness position while the backward propeller reduces the turbulence because of the induced negative pressure field. Also BLI propeller ingests more mass flow rate and apply less pressure ratio which is beneficial for the durability of the propeller.
BLI propulsion is favorable because it stabilizes the flow over the body and decrease power consumption.

References

[1] Leroy H Smith. Wake ingestion propulsion benefit. *Journal of Propulsion and Power*, 9(1):74–82, 1993.

[2] L. Teperin and A. Ujuhu. Method for determining pressure drag in aerodynamic interference problems. *TsAGI Science Journal. [in Russian]*, XXI(3):3–10, 1990.

[3] A.A. Razov. Numerical analysis of the efficiency of the location of the propeller in a viscous wake using Navier-

Stokes equations. *TsAGI Science Journal. [in Russian]*, XL(3):60, 2009.

[4] A.V. Kornushenko, O.V. Kudryavtsev, L.L. Teperin, L.N. Teperina, A.V. Shustov, F. Orfinejad, and M. Thein. useing the principle of useful interference to improve the aerodynamic perfection of the layout of the propeller and wing. *TsAGI Science Journal. [in Russian]*, XLVII(8), 2016.

[5] L.L. Teperin, L.N. Teperina, F.E. Orfinejad, M. El-Salamony, and M. Thein. Froude model of ideal propeller in inhomogeneous flow. *TsAGI Science Journal. [in Russian]*, XLVIII(3), 2017.

[6] Mark Drela. Power balance in aerodynamic flows. *AIAA journal*, 47(7):1761–1771, 2009.

[7] H Clyde McLemore. Wind-tunnel tests of a 1/20-scale airship model with stern propellers. Technical report, DTIC Document, 1962.

[8] F. Goldschmied. Aerodynamic design of low speed aircraft with a NASA fuselage/wake propeller configuration. AIAA-86-2693. In *AIAA/AHS/ASEE Aircraft Systems, Design and Technology Meeting*, pages 20–22, 1986.

[9] M Drela. Development of the D8 transport configuration, AIAA 2011_3970, AIAA appl. In *Aero. Conf*, 2011.

[10] Gérald Carrier, Olivier Atinault, Richard Grenon, and Christophe Verbecke. Numerical and experimental aerodynamic investigations of boundary layer ingestion for improving propulsion efficiency of future air transport. In *31st AIAA Applied Aerodynamics Conference*, page 2406, 2013.

[11] S.G. Ignatiev, L.N. Teperina, and M. El-Salamony. On the performance of propeller behind an airfoil at transonic speeds. *TsAGI Science Journal. [in Russian]*, XLVIII(3), pages 3–10, 2017.

[12] M. El-salamony and L. Teperin. 2D numerical investigation of boundary layer ingestion propulsor on airfoil. In *7th European Conference for Aeronautics and Space Sciences*, 2017.

[13] K. Sabo and M. Drela. Benefits of boundary layer ingestion propulsion. In *53rd AIAA Aerospace Sciences Meeting*, page 1667, 2015.

[14] P Lv, D Ragni, T Hartuc, L Veldhuis, and AG Rao. Experimental investigation of the flow mechanisms associated with a wake-ingesting propulsor. *AIAA Journal*, pages 1–11, 2016.

[15] J. Anderson. *Fundamentals of aerodynamics*. Tata McGraw-Hill Education, 2010.

Multidisciplinary optimization of a MAV propeller for noise reduction

F. Boyer†, A. Drapier†, Y. Mérillac†, C. Nana†* and Ronan Serre⋆
†Altran SO, 4, avenue Didier Daurat, 31700 Blagnac, FRANCE
⋆ISAE, 10, avenue Édouard Belin - BP 54032 - 31055 Toulouse Cedex 4, FRANCE

ABSTRACT

The following research aims at reducing the noise of a MAV's propeller without impacting its efficiency. To ensure reasonable computational time, low fidelity methods are used for each discipline: Lifting line for aerodynamics, beam model for structure and Ffowcs-Williams and Hawkings analogy for acoustics. Validation tests are performed to evaluate reliability of these methods on a reference configuration: a GWS80x45 propeller in hover. More complex CFD tools are also used at some points of the optimization process to check for the coherence and fidelity of the results. The open-source SU2 solver and a Lattice-Boltzmann Method solver serve this purpose. Finally, results are compared to an experimentation conducted by ISAE (Institut Supérieur de l'Aéronautique et de l'Espace) in an anechoic chamber.

1 INTRODUCTION

Micro-Unmanned Air Vehicules (MAV) are developing incredibly fast worldwide. They tend to be in every aspects of everyday life. Those are used for surveillance as well as in agriculture, firefights or delivery services. It is becoming crucial to reduce their noise. In that aspect, blade noise represents the most part of this disturbance. A part of this noise is due to pressure variations induced by aerodynamic around rotating blades, so it is possible to adapt blade geometry to reduce the noise. However, this geometric modification should not degrade performance of the blade: in an aerodynamic point of view, traction force must be kept constant, and efficiency of the blade should not decrease. At least two disciplines, aerodynamic and acoustics are involved, and a third discipline: mechanics, could be also considered to take into account deformation due to aerodynamic forces, and to ensure structural robustness, so a multidisciplinary optimization (MDO) is then necessary to solve this problem. Low-fidelity simulation tools are used for each discipline in order to have acceptable run times for the optimization. These disciplinary tools are integrated in GEMS[1] (Generic Engine for MDO Scenarios). GEMS is a platform developed by IRT St Exupery which allows performing multidisciplinary optimization. Thanks to the flexibility of this tool, a large number of MDO schemes can be tested and compared. This global process is integrated into OPTIMIND, an in-house workflow manager, to handle the connection between the different tools and to provide a user-friendly GUI. The GWS80x45 blade was selected for this as many experimental and computational data for this geometry were available. This allow validating the results provided by our low-fidelity tools before starting the optimizations. Furthermore, this blade offers room for improvement. The main goal here is to find the best methodology. For this reason both Mono and Multi-Disciplinary optimizations were launched to see the impact of each discipline on the solution.

*Email address(es): cyril.aymard.nana@gmail.com

2 THE MULTIDISPLINARY OPTIMISATION (MDO) METHODOLOGY

The Multidisciplinary Optimization (MDO) proposes solutions to design complex systems. When more than one discipline are involved in an optimization problem, it cannot be solved efficiently by performing sequential optimizations for each discipline. A more global approach combining all the disciplines must be adopted, taking into account coupling between disciplines. In this approach one global optimization algorithm is used to solve the whole problem. Different strategies, often called MDO formulations, exist: MDF, IDF and many other. The most common methods are MDF and IDF:

- MDF: Multi discipline feasible: equilibrium between disciplines is computed at each iteration of the optimization algorithm, the advantage of this method is that the computed solution is "feasible" all along the optimization process,
- IDF: Individual discipline feasible: equilibrium between disciplines is defined as a constraint of the optimization algorithm: iterations run faster but equilibrium between disciplines is obtained only at the convergence.

As usual in optimization, a "universal" method suited for any kind of problem does not exist. The best method depends on the problem studied. In the end three points are crucial to efficiently lead a MDO study:

- Definition of the optimization problem: objective(s), constraints,

- Optimization algorithm and MDO strategy used,
- Workflow management: communication between disciplinary tools, and integration in the optimization loop.

GEMS is used for the first two items. It offers a large choice of optimization algorithms and MDO formulations. It is a very flexible tool which allows testing several problem formulations easily. The in-house tool OPTIMIND is used for workflow management. It allows wrapping and linking disciplinary tools easily in order to integrate them in the optimization loop handled by GEMS.

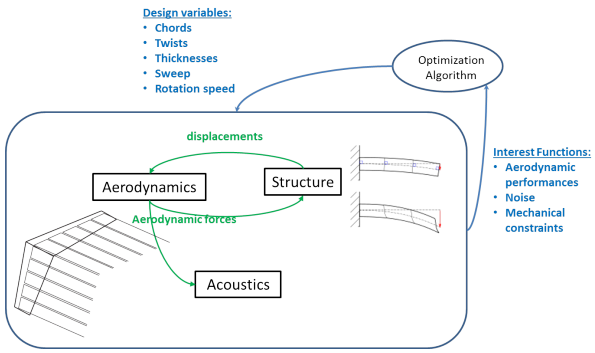


Figure 1: The Multidiscplinary Optimization procedure.

2.1 *Aerodynamic solver:* QuickCFD.py

QuickCFD computes aerodynamic loads and performance of lifting surfaces with complex geometrical properties including arbitrary camber, sweep, dihedral and twist. It relies on an adaptation of the general numerical lifting-line method based on a fully three-dimensional vortex lifting law developed by Phillips and Snyder [2]. The accuracy of this method was shown to be comparable to that obtained from panel methods at a small fraction of the computational cost.

2.2 *Structure solver:* MECHA-BLADE

MECHA-BLADE is an in-house program developed to carry out mechanical calculations on turbomachinery blades. The code is based on the variational formulation approach proposed by Rao [3, 4] which takes into account pre-twisted cantilever beams with an asymmetric airfoil cross-section mounted at a stagger angle on a rotating disc. The use of this semi-analytical approach aims at providing fast and relatively accurate calculations, particularly in an optimization context or in preliminary design stages where a large amount of sensitivity analyses for several parameters are required. In practice, the blade is discretized into several cross sections from which the mass and stiffness matrices are calculated. The natural frequencies and mode shapes are then determined using an eigenvalue extraction routine. A modal approach is then used to complete the structural analysis and determine the displacement and stress response of the blade subjected to both centrifugal and aerodynamic forces. The figure 2 shows the model flow chart.

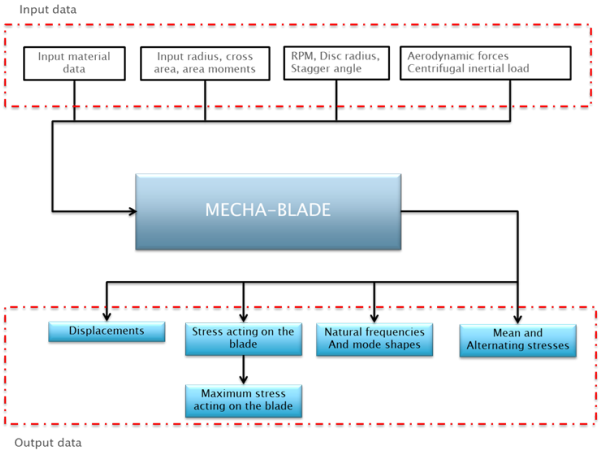


Figure 2: Model flow chart of the MECHABLADE program.

2.3 *Ffowcs-Williams & Hawkings solver:* pyFfonc.py

The acoustic propagation is done through the pyFfonc.py program. It solves the Ffowcs-Williams and Hawkings aeroacoustic analogy using Farassat's 1A formulation[5] as described by Casalino[6]. The inputs are the drag, lift and induced velocity on a blade of the propeller, the latter being divided into several panels. The acoustic power is then computed by summing the radiated noise all around the propeller.

3 Validation of the method

3.1 *QuickCFD validation*



Figure 3: The GWS80x45 reference propeller.

Since many experimental and computational data are available for GWS80x45 geometry, this configuration was used to validate QuickCFD results. The propeller is represented in Figure 3. Figure 4 compares Drag and Lift forces repartition along blade radius with the results of a Blade Element (BEMT) computation performed at ISAE. QuickCFD seems to be quite accurate to compute lift forces. However the computed drag is inferior to the BEMT value. This can be explained by the fact that only induced drag component is computed by QuickCFD, friction or viscous pressure drag

are not taken into account. In order to validate this hypothesis and to ensure that the induced drag computed by QuickCFD is correct, a comparison of induced velocity was done. The results are presented in Figure 5. A good agreement between BEMT data and quickCFD results is reached.

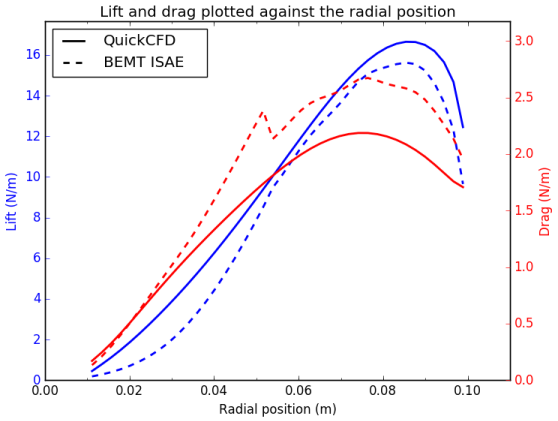


Figure 4: Comparison of QuickCFD results with experimental data for GWS80x45

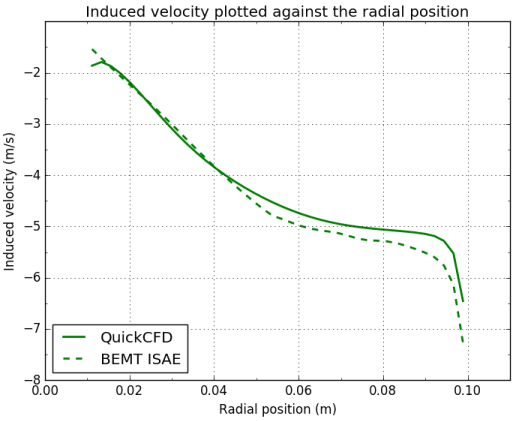


Figure 5: Comparision of induced velocity computed by QuickCFD with experimental data

The evolution of the lift force with respect to the rotation speed of the propeller was also compared to existing results: experimentations or computations with STAR CCM+, LBM methods (PowerFlow) or BEMT (Blade Element Momentum Theory). The propeller is composed of two blades as shown in Figure 3 even if the QuickCFD computation only considers one blade. A summary of the results is given by Figure 6, QuickCFD results are conform with the reference data. To conclude, these validation tests demonstrate that QuickCFD is able to compute aerodynamic forces with good accuracy, and can be used for optimization studies.

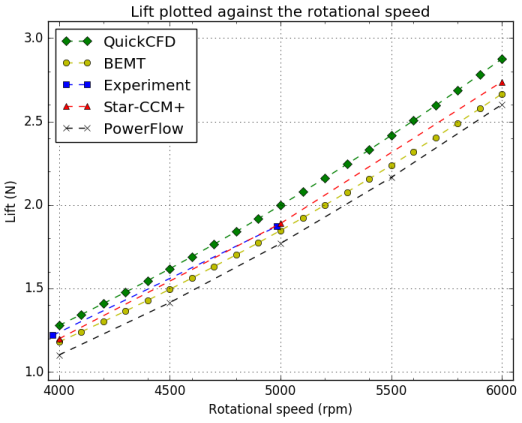


Figure 6: Evolution of lift with rotation speed

4 Optimization Study

The design parameters for this study are the geometrical parameters of the blade: chord length, sweep, dihedral and chord values along blade radius. Spline curves are used to control the evolution of these geometric laws so as to obtain smooth geometries. The number of control points used to define these splines is chosen by the user. More control points offers more flexibility to explore different shapes but it also involves slower convergence for the optimization. The rotation speed of the blade is also controlled during the optimizations to adjust traction level.

This study is divided in three steps in order to understand well the influence of parameters on each discipline and the influence of discipline coupling on the optimization process. These steps are:

1. Mono disciplinary (Aerodynamic) optimization: The goal of this study is to maximize the efficiency of the blade in hover with a constraint of minimal thrust to respect. This first study shows what are the most influent parameters on the aerodynamics of the blade and the result of this optimization provides an ”ideal” goal for the two following optimizations.

2. Aero-acoustic optimization: For this optimization, the noise generated by the blade is taken into account. Two strategies can be investigated: a bi-objective optimization can be performed or noise can be treated as a constraint of the optimization algorithm. Comparing results obtained with those of the pure aerodynamic optimization helps understanding the influence of design parameters on both disciplines.

3. Full MDO study: Aerodynamic- Acoustic-Structure: The goal of this final study is to add structure discipline in the process with a coupling between aerodynamic and structure by considering the blade displace-

ment induced by aerodynamic loads. This part of the study was not done yet due to a lack of time but it is necessary and will be performed in the future.

Results of the three studies are analyzed and compared to see the benefit of a multi-disciplinary approach for our problem.

4.1 Aerodynamic optimization

- Objective: **minimize Drag force (N)**
- Constraint: **maintain Lift force up to 2.08 N** (Reference value for GWS80x45 at 5000 rpm)
- Design variables: 7 control points were used to define spline curves controlling Chord, Twist, Dihedral and Seewp evolution along blade radius: **28 design variables**
- Algorithm: In order to have a fast convergence, we chose a gradient-based algorithm: **SLSQP**, available in GEMS, allows performing gradient based optimization, with one ore several constraints. GEMS also provides a finite differences module to compute aerodynamic gradients.

The aerodynamic optimization converges in 30 iterations as shown by Figure 7. The comparison of the aerodynamic forces (computed by QuickCFD) between baseline and optimized blade is given in Table 1: Drag force decreases by 28%, Lift force is kept at the same level. Figure 8 compares baseline and optimized geometries: as we can see globally sweep and mean chord decreased, whereas Twist increased near the root of the blade. These geometric modifications involved an increase of the rotation speed to keep the Lift force constant.

QuickCFD	Baseline	Optimized	delta (%)
Drag (N)	0.3264	0.2350	**-28.02**
Lift (N)	2.087	2.087	**2.26e-7**
Torque (N/m)	0,0177	0,0129	**-27.34**
Rot. speed (rpm)	5000.0	6206.12	**24.12**

Table 1: Aerodynamic optimization summary

In order to validate results of this aerodynamic optimization, calculations were done on baseline and optimized geometries using PowerFlow: surface repartition of aerodynamic forces is represented by Figure 10. Table 2 presents results for Torque and Lift: PowerFlow gives almost the same results than QuickCFD for Lift, Torque levels computed by PowerFlow are higher but we obtain the same relative difference between baseline and optimized blade than in QuickCFD, which confirms better performances of the optimized shape, and enforces reliability of this first aerodynamic optimization results

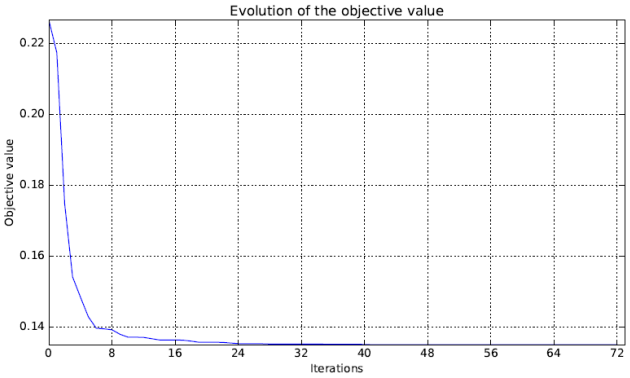


Figure 7: Aerodynamic optimization convergence history

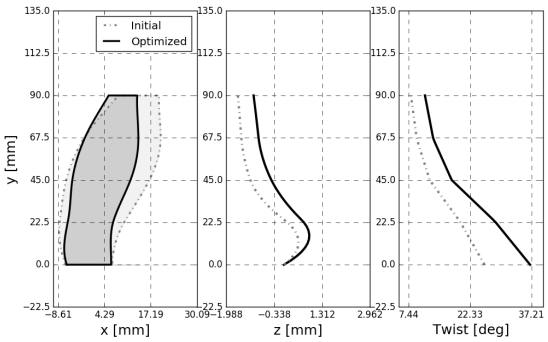


Figure 8: Aerodynamic optimized geometry: left: XY planform, center: Dihedral, right: Twist

4.2 Aeroacoustic optimization

- Objective: **minimize weighted sum of Torque and Acoustic Power**:

$$Obj = w * \frac{T}{T_0} + (1 - w) * \frac{Pa}{Pa_0} \qquad (1)$$

- Constraint: **maintain Lift force up to 2.08 N** (Reference value for GWS80x45 at 5000 rpm)
- Design variables: same parameters than Arodynamic optimization: **28 design variables**
- Same algorithm than the one used for Aerodynamic optimization: **SLSQP**

Aero-Acoustic optimization converges in 10 iterations as shown by Figure 11, comparison of aerodynamic forces

PowerFlow	Baseline	Optimized	delta (%)
Torque (N/m)	0,0426	0,0319	**-25.01**
Lift (N)	2,0276	2,1249	**4.8**

Table 2: Aerodynamic optimization: PowerFlow results

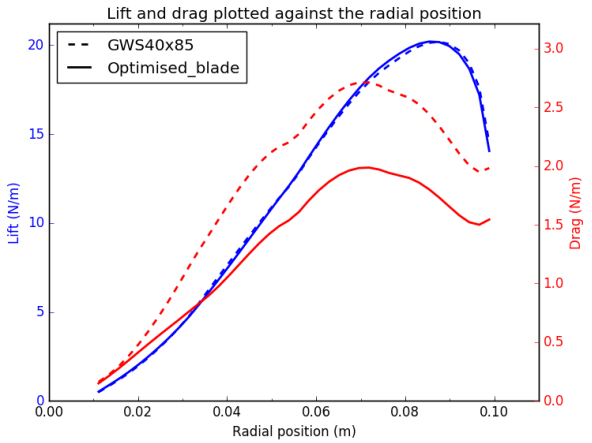


Figure 9: Aerodynamic optimization: radial forces comparison

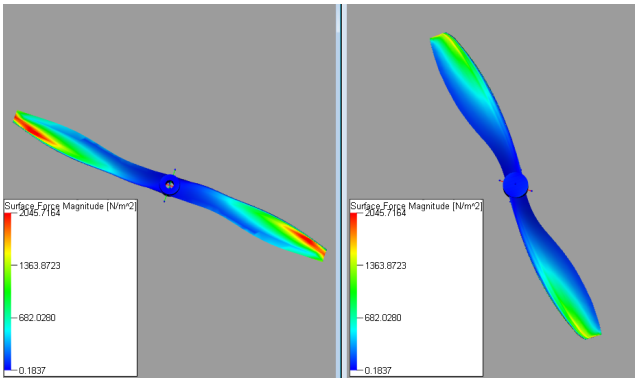


Figure 10: Aerodynamic forces computed by PowerFlow: left: Optimized, right: Baseline

(QuickCFD) and noise levels (PyFfonc) between baseline and optimized blade is given in Table 3: Similarly to pure Aerodynamic optimization, Drag force decreases by 28%, Lift force is kept almost at the same level. Figure 12 compares baseline and optimized geometries: the optimized geometry seems quite similar to the one obtained by Aerodynamic optimisation: both shapes are compared in Figure 14: the only differences are the twist and chord distribution near blade root. Optimized blade reduces the noise by **1dBA**, which represent a reduction of **20.41%** of the acoustic power.

4.3 Full multdisplinary optimization

Due to a lack of time this study could not been performed yet. Nevertheless, a fonctionnal structural tool was developed: MECHA-BLADE. It will soon be integrated in GEMS to perform aeroelastic computations. The previous two optimisation studies (Aerodynamic and Aero-Acoustic) will then be performed with MECHA-BLADE-QuickCFD coupling, to see its impact of structure on optimization results. In a second time optimization problem are to be improved to take into accound structural aspects: to limit vibrations or structural constraints level in the blade for instance.

QuickCFD-PyFfonc	Baseline	Optimized	delta (%)
Drag (N)	0.3224	0.2342	**-27.1**
Lift (N)	2.057	2.0574	**-0.01**
Torque (N/m)	0,0176	0,0126	**-27.84**
Rot. speed (rpm)	5000.0	6196.12	**23.92**
Noise (dBA)	67.28	66.28	

Table 3: Aero-Acoustic optimization summary

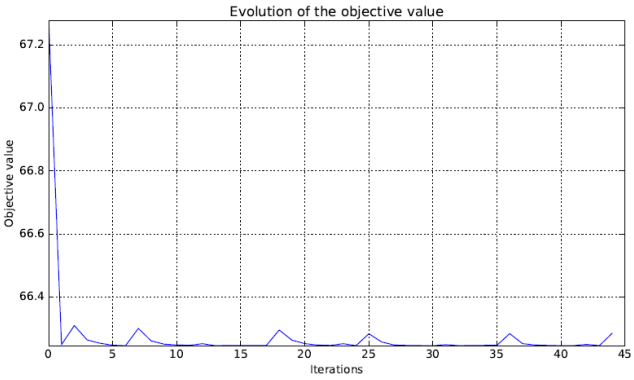


Figure 11: Aero-Acoustic optimization convergence history

5 Conclusion

A multi-disciplinary/multi-objective method for the aeroacoustic optimization of MAV propeller's blade has been developed in this study. It was tested on a well known propeller, the GWS80x45. First aeroacoustic optimizations showed some very interesting results such as a decrease of the drag force on the propeller of about 27% for an acoustic radation reduced by 1dB. A full aero-structure-acoustics optimisation has still yet to be performed. The new blade design

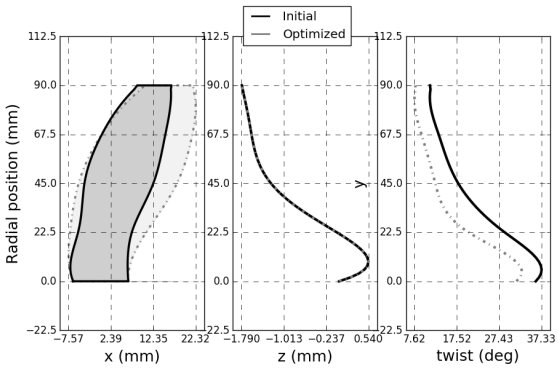


Figure 12: Aero-Acoustic optimized geometry: left: XY planform, center: Dihedral, right: Twist

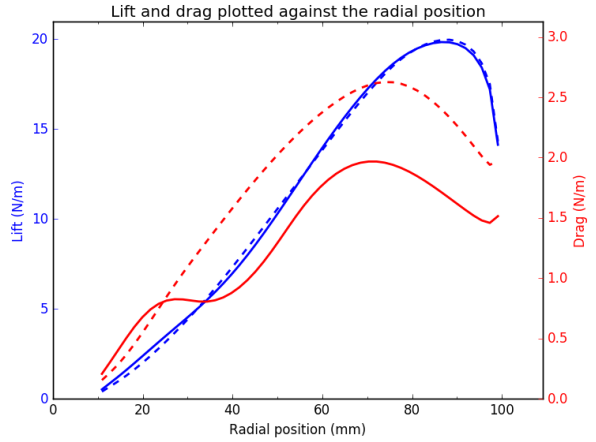


Figure 13: Aero-Acoustic optimization: radial forces comparison

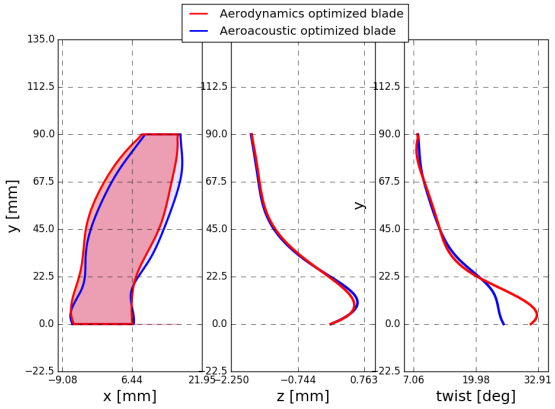


Figure 14: Comparison between Aerodynamic and Aero-Acoustic optimized geometries

will then be 3D-printed and tested in an anechoic chamber to confront with the experiment.

Acknowledgements

Authors would thank the ISAE and Ronan Serre for the experiments, Maxime Itasse and Said Ouhamou for Quick-CFD and MECHABLADE respectively. Authors also thank the IRT St Exupery for the GEMS platform and Romain Denis from EXA for the PowerFlow LBM solver.

References

[1] F. Gallard A. Gazaix and V. Gachelin. Towards the industrialization of new mdo methodologies and tools for aircraft design. *(to appear) AIAA conference*, 2017.

[2] W. F. Phillips and D. O. Snyder. Modern adaptation of prandtls classic lifting-line theory. *AIAA Journal*, 37(4):662–670, 2000.

[3] J. S. Rao. *Turbomachine Blade Vibration*. New Age International, 1991.

[4] J. S. Rao and N. S. Vyas. Application of Reissner method to free vibrations of a tapered, twisted, aerofoil cross-section turbine blade, mounted at a stagger angle on a rotating disc. *Def Sci J.*, 36(3):273–292, 1986.

[5] F. Farassat. Linear acoustic formulas for calculation of rotating blade noise. *AIAA J.*, 19(9):1122–1130, 1981.

[6] D. Casalino. An advanced time approach for acoustic analogy predictions. *J. Sound Vib.*, 261:583–612, 2003.

[7] J. L. Martins. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51(9):2049–2075, 2013.

[8] J. Clément. *Optimisation multidisciplinaire : étude théorique et application la conception des avions en phase d'avant projet*. PhD thesis, Université de Toulouse, ISAE, 2009.

Propeller Performance Calculation for Multicopter Aircraft at Forward Flight Conditions and Validation with Wind Tunnel Measurements

C. Molter*, P.W. Cheng,
University of Stuttgart, Wind Energy Research Group (SWE) at the Institute of Aircraft Design
Allmandring 5b, 70569 Stuttgart, Germany

ABSTRACT

When designing a fast flying multicopter aircraft knowledge about propellers at inclined inflow conditions is important. To investigate this operating condition a whirl tower was built and several propellers were tested in a wind tunnel at angles of attack of $8°$, $15°$, $22.5°$ and $30°$. The inflow speeds were varied between $4\ m/s$ and $30\ m/s$.

The gained measurement data was used to validate an in-house blade element simulation software. The simulations were improved by adjusting airfoil lift and drag polars to static propeller measurements. Without any further adjustments to the inclined inflow condition the simulations showed good agreement with the measurements of two different propellers operating at an angle of attack. This means that it is possible for future projects to gain significant knowledge about propellers at an angle of attack with the use of static thrust and performance measurements without the need for a wind tunnel.

A less complicated semi-analytical approach was also tested to model the performance of propellers at an angle of attack. Without further adjustments to the equations it was not possible to achieve a good agreement with measurement data using the simplified approach.

Some measurements were also taken with a counter-rotating propeller arrangement (coaxial rotors). A hypothesis is proposed that the thrust deficit of the bottom propeller due to the influence of the top propeller is less at forward flight conditions than at the static operating condition. This hypothesis could be confirmed by measurements but still needs further validation.

1 INTRODUCTION

In the scope of the ANWIND project a multicopter aircraft for wind measurements near wind turbines will be designed, built and tested at the University of Stuttgart's Wind Energy Research Group (SWE). The two main challenges in the development process of this aircraft, called ANDroMeDA (ANWIND Drone for Measurement and Data Acquisition) are a reasonable performance in terms of stable flight at windy conditions and long flight duration as well as placing the measurement probe away from disturbances of the propellers.

Since ANDroMeDA will be hovering at wind speeds of $11\ m/s$ ((the design wind speed of a wind turbine) and above it is important to have knowledge on how the propellers will perform at an angle of attack and at higher inflow speeds. Without experiences on thrust and power of inclined propellers it is even impossible to make a statement about the resulting flight times.

For static propeller operation or propellers with a straight inflow (airplane operation) several open source simulation tools exist e.g. Qprop, JavaProp, JBlade [1, 2, 3] and some measurement data is also available [4]. Hence it is more or less easy to calculate the hover performance of a multicopter rotor in the absence of wind.

For a multicopter that is designed to be able to hover in strong headwinds or a multicopter designed to be most efficient at higher forward flight speeds the available simulation tools cannot be used and propeller measurements are really rare. Larger helicopter companies have their own in-house blade element simulation tools. There are also commercial tools available for example CAMRAD [5]. Since those commercial tools are above the budget of most UAV projects and are capable of much more than needed for this task the UAV department at the SWE developed a small custom made blade element tool, called *RotoCalc*, which is able to simulate inclined propellers.

Since *RotoCalc* could not be validated with measurement data in the past and since no wind tunnel measurements for inclined propellers are available for the desired propeller sizes a propeller test rig, called whirl tower, was built and several propellers have been tested at inclined inflow conditions. Two propellers, an APC Thin Electric 13x6.5 and an Aeronaut CAM Carbon Light 13x6 have been simulated. Because a counter-rotating propeller configuration is also considered for the ANDroMeDA multicopter to achieve a higher redundancy another test rig was built for this propeller arrangement additionally.

*Email address: molter@ifb.uni-stuttgart.de

2 Description of the Test Setup and Simulation Environment

2.1 Whirl Tower Setup

The whirl tower is shown in Figure 1 and Figure 2. It measures propeller thrust, propeller torque, propeller RPM, battery voltage, motor current,motor temperature and the current ESC input signal. All measurements are transmitted to a PC at a rate of $50\ Hz$. Thrust and torque measurements are taken using low cost load cells [6] and low cost load cell amplifiers [7]. To make sure no aerodynamic forces will disturb the load cells the whirl tower is covered with a clear plastic cylinder. The RPM is determined electronically with the help of one phase of the brushless motor.
All electronics is placed on a single circuit board included in the whirl tower to keep cable lengths short. The whirl tower is connected to the data acquisition laptop via bluetooth. This ensures a galvanic isolation. The PC software (Figure 3) displays all measurements in real time. It is also possible to send commands e.g. for taring the load cells and to control the motor power with this software.
To calibrate the torque and force measurements a special arrangement of pulleys and wires has been designed shown in Figure 2. The friction of the pulleys at loaded condition could be determined to be less than $3\ g$. During calibration the maximum relative error in thrust was about 0.6% and the maximum relative error in torque was about 1.6%. For most load conditions these errors were much smaller.
The RPM measurement quality is hard to characterize since it is not known if deviations are originated in the measurement or in the ESC and motor. After comparison with a strobe lamp the measurement accuracy is assumed to be about ± 10 RPM. Another source for measurement errors is the whirl tower itself and its aerodynamic wake which is of course larger than in the usual multicopter arrangement with a thin motor arm.

2.2 Counter-Rotating Propeller Test Setup

For the counter-rotating propellers another test setup was used. The thrust of the upper and lower propeller is measured separately with the same kind of load cells used for the whirl tower [6]. The RPM of the upper and lower propeller as well as the motor current of upper and lower motor is also measured separately. Additionally the battery voltage is measured. Figure 4 shows the test setup.
Unfortunately the mounting stiffness of the connection between load cells and motors was too low so that oscillations occurred. These oscillations only occurred with the wind tunnel running. All static tests were not influenced by oscillations. Since the wind tunnel time was limited and the problem was not revealed before the campaign there was no time to improve the motor mounts. The load cell stiffness itself is much higher than the stiffness of the motor mounts so that thicker motor mounts, also resulting in a slightly higher propeller distance, should eliminate this problem for future measurements. The distance of the two propeller is

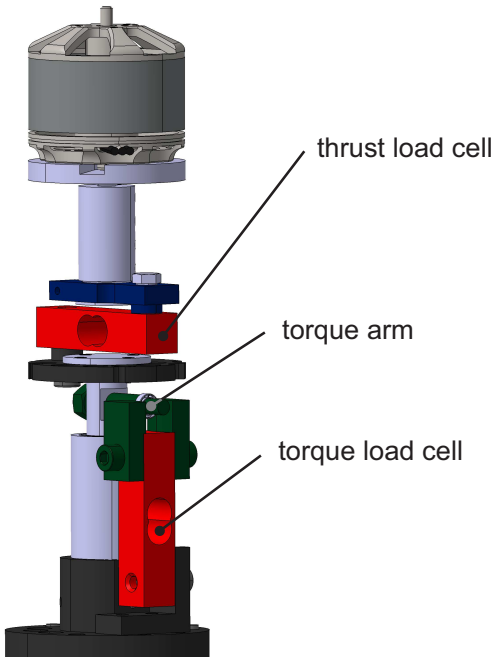


Figure 1: Load cell arrangement at the whirl tower.

$70\ mm$ from blade tip to blade tip in the current setup.

2.3 Wind-Tunnel Setup

The experiments described in this article were conducted in the medium size low speed wind tunnel at the Institute Aerodynamics and Gas Dynamics (IAG) at the University of Stuttgart. This wind tunnel is a closed Gttinger type tunnel with an open jet test section. The nozzle used has a diameter of $1.0\ m$. Since the whirl tower and counter-rotating propeller test rig are working completely independent from all other equipment no further measurements were needed from

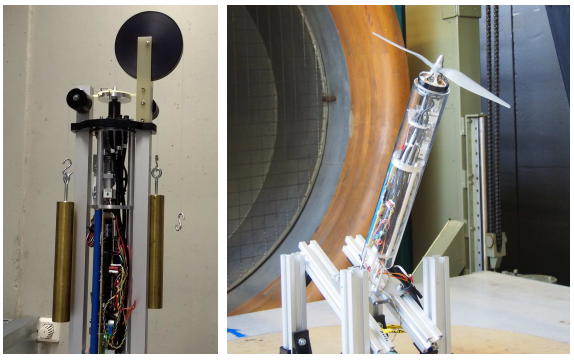


Figure 2: Left: Calibration Unit with a pair of torque calibration weights applied; Right: Whirl tower mounted in the wind tunnel and tilted forward.

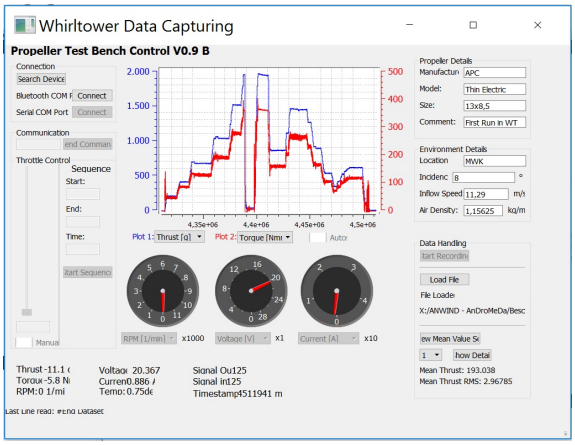


Figure 3: Software to control the whirl tower and save measurements.

the wind tunnel except the wind speed measurement. The wind speed measurement is accomplished with a pitot tube, a static pressure hole and very accurate humidity and temperature sensors. Since the whirl tower measurements were averaged over time during post processing and since the wind tunnel speed fluctuates only very little the time averaged wind speed was transferred from hand to the whirl tower control software and saved with each measurement file.

2.4 *Simulation Software RotoCalc and Simulation Settings*

The Software *RotoCalc* uses the blade element method as described for example in [8] and [9]. Since this procedure is a pretty standard approach a more detailed description is not part of this article and only the relevant settings are described. The induced velocity can be determined in three different ways with *RotoCalc*. Two of them were used for this investigation. For the static case the well-known Blade Element Combined Momentum Method (also called BEMT) was used. This method determines an induced velocity in a way that each annular ring of the rotor disc will produce the same thrust whether calculated by simple momentum theory or by two dimensional airfoil theory. More information can be found for example in [8] or [9]. The method will result in a non-uniform induced velocity along the blade radius.
RotoCalc is also able to use the BEMT method for an inclined propeller. But it appeared that this results in a less accurate agreement with wind tunnel measurements. For the inclined propeller operation a uniform velocity along the rotor disc resulted in a better agreement with measurement data. The induced velocity v_i at the propeller disc for a propeller inclined by the angle α_{prop} and operated at the thrust T while experiencing an inflow speed V_∞ is determined by:

$$v_i = \frac{T}{2\rho A_{prop} V_{res}} \quad (1)$$

$$V_{res} = \sqrt{(sin(\alpha_{prop})V_\infty + v_i)^2 + (cos(\alpha_{prop})V_\infty)^2} \quad (2)$$

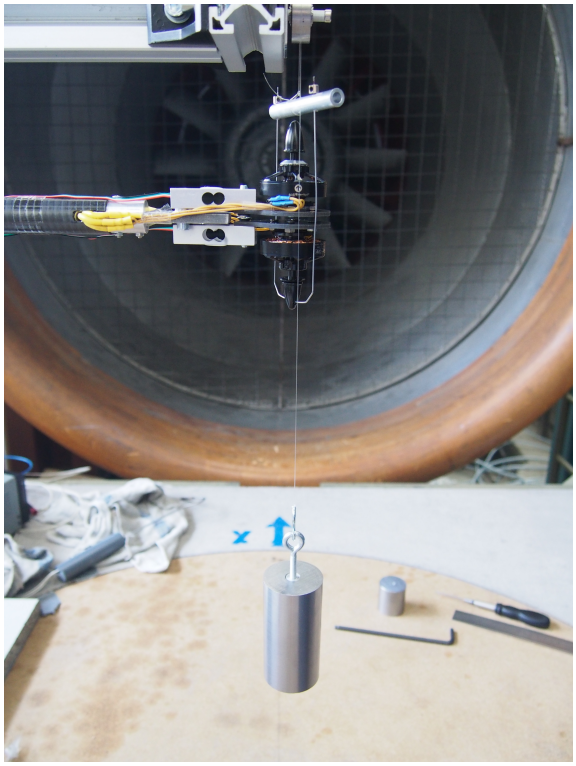


Figure 4: Counter-rotating propeller test setup with calibration weight for the bottom load cell applied.

Where A_{prop} is the propeller disc area and ρ is the air density. This semi-empirical equation was first formulated by Glauert [10]. The equation cannot be solved analytically. *RotoCalc* solves this equation numerically after each full revolution using the false position method. This is done till the thrust evaluated by Equation 1 and the the thrust evaluated by two-dimensional airfoil theory / blade element theory converge.
Tip-losses are modeled quite simple with Prandtl's tip-loss factor as described for example in [9]. Therefore the blade element thrust is neglected at the thrust integration after a certain radial position. This radial position is called the effective blade radius R_e. It is decreased by the factor B. $B = 0.95$ was used for this simulations.
The propeller blades can be modeled in *RotoCalc* by several radial positions. For each position blade chord and twist as well as the airfoil to use can be set. Between those radial positons *RotoCalc* will interpolate linearly. The number of blade elements to use for the interpolation can be set to an arbitrary number. For the APC propeller an overall number of 96 radial elements was used and for the Aeronaut Propeller an overall number of 99 radial elements was used. A finer resolution did not show any change of the simulations results.
Since the inclined propeller does not experience a rotational symmetric inflow it also has to be discretized in the azimuth

direction. The azimuth step size can be set freely with *RotoCalc*. For both simulations 72 azimuth steps ($\Delta\Psi = 5°$) have been used. A finer resolution did not change the results.
In-plane or out-of-plane bending as well as twisting of the propeller was not considered. The propeller was modeled as a totally rigid propeller without flapping or lagging. In *RotoCalc* different airfoils and hence different lift and drag polars can be set up. If two different airfoils are set for two following radial positions both airfoils will be taken into account and *RotoCalc* will interpolate between the two resulting lift and drag coefficients of both airfoils.
The airfoils are specified by look up tables with an arbitrary number of angles of attack. To account for Reynolds number effects several lift and drag polars can be set at several Reynolds numbers for each airfoil .
Instead of a look up table an equations can also be used to model lift and drag. This was used to adjust the polars to a better agreement with static measurements as described below. The angle of attack for a propeller can be set between $\alpha_{prop} = 0°$ (helicopter rotor in hover or climb / airplane propeller) and $\alpha_{prop} = 90°$ (helicopter rotor with parallel inflow).

3 Static Propeller Measurements

As preparation for the wind tunnel measurements a number of 18 two-bladed propellers and two three-bladed propellers between 13 inch and 14 inch diameter have been tested at static operation on the whirl tower. A complete list of the results will not be given here. On one hand it is beyond the scope of this article to give an extensive propeller market overview. On the other hand it is really hard to tell which propeller is the best regarding the maximum thrust at a given power or the highest figure of merit for a given thrust. It can be said that the aerodynamic performance of most of the propellers was not differing very much and differences were often located in the range of measurement uncertainties. Because the propellers are very similar regarding the aerodynamic performance other factors like price and weight start to play a more important role.
Two general remarks can be given: Wooden propellers had a slightly lower figure of merit than those made of plastics. This can be explained by their higher relative airfoil thickness resulting from the lower structural strength of the material. Expensive carbon fiber propellers could not proof to have a better aerodynamic performance. At least not in the range of the accuracy given by the whirl tower. But these propellers have of course a lower weight which will result in a higher weight remaining for the flight battery and hence in an increased overall flight performance.

4 Propeller Modeling

Two different propellers were modeled for BE simulations: An APC Thin Electric 13x6 and an Aeronaut CAM Carbon Light 13x6 propeller. The geometric properties were determined purely manual using a caliper, angle templates and cutting the propeller at several blade positions to estimate the used airfoils / proper airfoils to estimate the propeller performance. A more sophisticated method, e.g. involving a 3D scanner, could be used in the future of course to decrease the remaining geometric uncertainties.
After determining airfoils with suitable thickness and camber lift and drag polars between $\alpha = -30°$ and $\alpha = +30°$ have been determined at several Reynolds numbers with the help of the two-dimensional panel method simulation software XFOIL [11]. To be able to adapt the simulations to measurements by fine-tuning the polars, the XFOIL results were then described by the following analytical equations:

$$c_l(\alpha) = cl0 + cla \cdot \alpha \tag{3}$$

$$c_d(c_l) = cd0 + cd2(c_l - clcd0)^2 \tag{4}$$

$$cd2 = cd2u \;\; if \;\; c_l < clcd0 \tag{5}$$

$$cd2 = cd2l \;\; if \;\; c_l < clcd0 \tag{6}$$

By using these coefficients it is possible to adjust a polar towards higher/lower lift and higher/lower drag without editing every single line in the polar lookup table. Modeling lift- and drag-polars with these coefficients can also be found in QProp [1] and therefore additionally results in the possibility to compare static *RotoCalc* simulations to QProp simulations easily.

5 Fine-Tuning of Simulations

Figure 5 and figure 6 show the measurement results for the APC and Aeronaut propeller in comparison with the initial and the tuned simulations. The initial simulations have been done without any further adjustments to measurement data. Propeller chord and twist were modeled according to geometric measurements and the initial lift and drag polars from XFOIL [11] runs have been used.
Those initial simulations not only suffer from uncertainties which airfoil is really used for the actual propeller but also the flow at a rotating propeller can be quite complex. It includes three-dimensional effects like the effect of centrifugal force on the blade's boundary layer, low Reynolds number effects as well as additional turbulence introduced to the boundary layer by the rotating propeller. Some observations and a comparison between BE simulations, three-dimensional CFD simulations and an experiment can be found in [12]. Because of this circumstances without measurement data the initial simulations can only be seen as a first guess. To the author's knowledge it is also quite common for the development of manned helicopters to adjust BE simulations to whirl tower measurements by fine-tuning lift and drag polars.
By manually adjusting the lift and drag polars at three different Reynolds numbers, corresponding propeller operation at $n = 2000\ RPM$, $n = 4500\ RPM$ and $n = 7000\ RPM$, a very good match between simulation and measurement could be achieved for the static propeller operation. It has to be em-

phasized that all this steps can be done completely without the necessity of wind tunnel measurements.

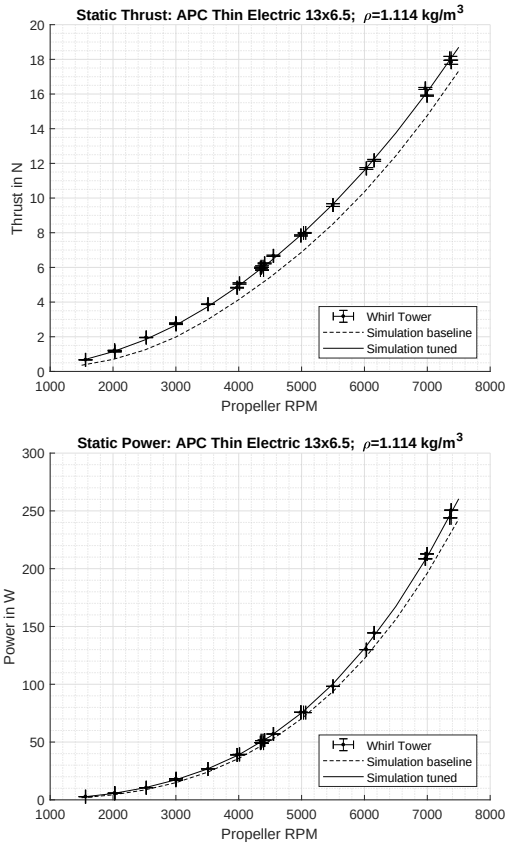


Figure 5: Static measurements for APC propeller in comparison to the initial and the tuned BE simulation.

6 Results for Propellers at an Angle of Attack

Table 1 shows the angles of attack and inflow speeds of all measurements taken. The inflow speeds were adjusted to cover the steady flight operation of a multicopter.

α_{prop} in °	8	15	22.5	30
V_∞ in m/s	8;11	8;11;15	12;15;20	17;20;30

Table 1: Range of the investigated operating conditions.

6.1 Propeller Performance at ANDroMeDA's Design Operating Point

At $\alpha_{prop} = 8°$ and $\alpha_{prop} = 15°$ measurements as well as simulations demonstrate, that a slightly higher thrust can be achieved with nearly the same or only a little more power. This can be seen in Figure 7. This fact is very satisfying since

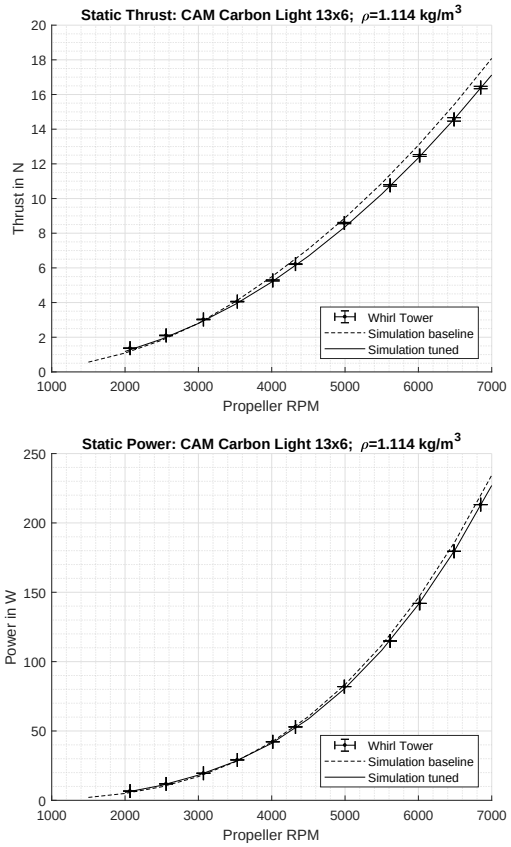


Figure 6: Static measurements for Aeronaut propeller in comparison to the initial and the tuned BE simulation.

this range of propeller angles of attack corresponds with ANDroMeDAs design operating point. Even with simple momentum theory the additional gain in thrust at the same power can be observed. Equation 1 can be used to calculate the induced velocity for an inclined propeller and hence the induced power needed for a certain thrust T:

$$P_{ind} = T \cdot (V_\infty \cdot sin(\alpha_{prop}) + v_i) \tag{7}$$

Using equation 1 and equation 7 the induced power for $\alpha_{prop} = 15°$ and $\alpha_{prop} = 30°$ at different inflow speeds can be calculated. A reduction in power can be seen for the lower angles of attack while at higher angles of attack the induced power will always be higher than in static operation (Figure 8). It has to be taken into consideration that for very low angles of attack ($\alpha_{prop} < 10°$) the reduction of induced power will be greatly overestimated by Equation 1.

6.2 Agreement of Simulations with Measurements

In general the *RotoCalc* simulations agree very well with wind tunnel measurements. Figure 9 depicts the worst fit at $\alpha_{prop} = 8°$ and the best fit at $\alpha_{prop} = 30°$ in case of the APC propeller. The worst agreement is a deviation from simulated

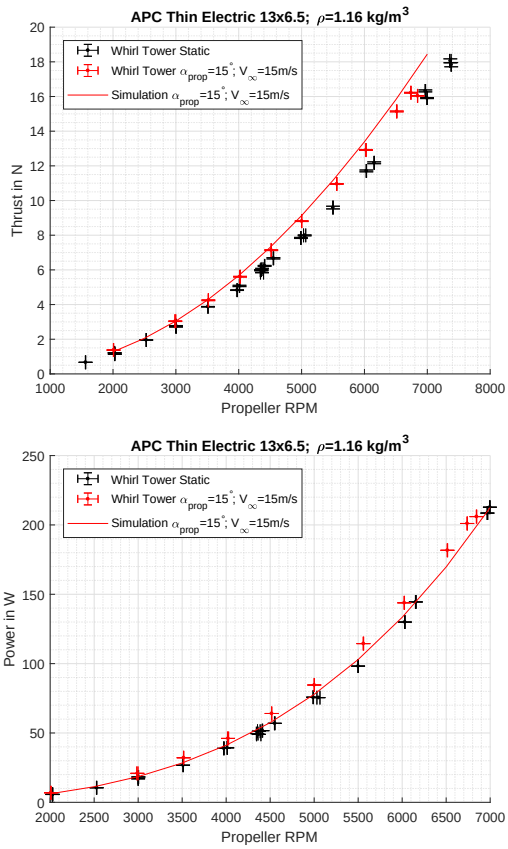


Figure 7: Thrust and power at $\alpha_{prop} = 15°$ compared to static operation.

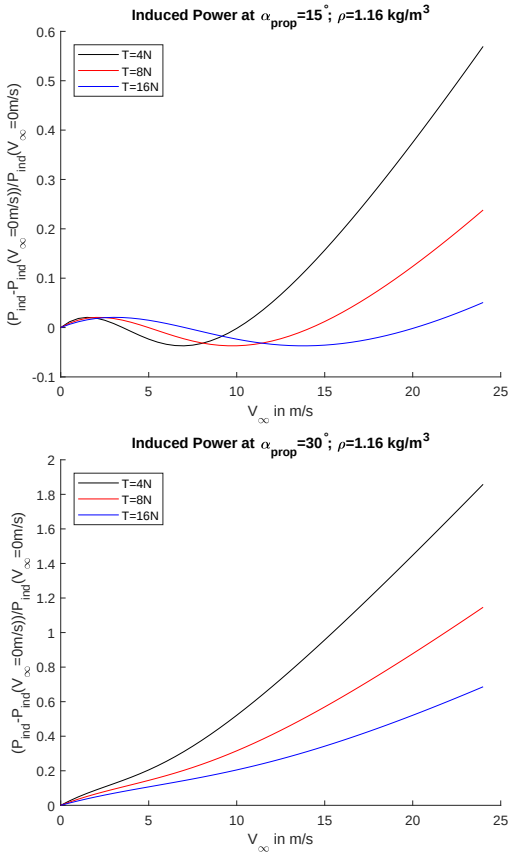


Figure 8: Induced power according to Equation 1 at different angles of attack.

power of 9.7 % and 11, 4 % from simulated thrust at $\alpha_{prop} = 8°$ at $V_\infty = 11m/s$. In most operating points the difference between measurements and simulation is much smaller.

7 Analytical Model for Inclined Propellers

A less computational and less coding intensive approach to model the performance of inclined propellers is given in [8]:

$$C_T = 0.5 \cdot cl_a \cdot \sigma \cdot (\theta_{075}/3 + 0.5 \cdot \theta_{075} \cdot \mu^2 - \lambda/2) \quad (8)$$

$$C_P = C_T^2 \cdot \kappa/(2\mu) + cd0 \cdot \sigma/8 \cdot (1 + 4.6 \cdot \mu^2) \quad (9)$$

Where $\mu = V_\infty \cdot cos(\alpha_{prop})/(\Omega r)$ is the advance ratio and $\lambda = (V_\infty \cdot sin(\alpha_{prop}) + v_i)/(\Omega r)$ is the inflow ratio. cl_a is the slope of the blade's airfoil lift polar and σ is the solidity of the rotor / propeller. θ_{075} is the blade's pitch at $r = 0.75 \cdot R$ and $cd0$ is the airfoil's zero lift drag. κ is an empirical factor accounting for tip-losses, losses of non-uniform inflow and other losses. Instead of thrust and power the dimensionless coefficients $C_T = T/(\rho \cdot A \cdot (\Omega R)^2)$ and $C_P = P/(\rho \cdot A \cdot (\Omega R)^3)$ are used.
Since the induced velocity v_i depends on the thrust from equation 1 the above expression cannot be solved analytically and the problem has to be solved numerically in an iterative manner.
This simple approach leads to a reasonable agreement of thrust at low angles of attack when the difference to the static case is low. For higher angles of attack the agreement is unfortunately of poor quality and the thrust will be significantly underestimated (Figure 10). The power was in general underestimated by this method for the investigated cases.

8 Some Observations on Counter-Rotating Propeller Configurations

In the static case the maximum thrust of the upper propeller was not influenced significantly by the operation of the lower propeller. While the thrust changed by less than 2% the power at full throttle was increased by 6%.The bottom propeller on the other hand was severely influenced by the operation of the top propeller. Maximum thrust of the bottom propeller was decreased by 30% and the power of the lower propeller was decreased by 10%.
Some theoretical background on static counter-rotating propeller operation can be found in [13]. A very simple approach

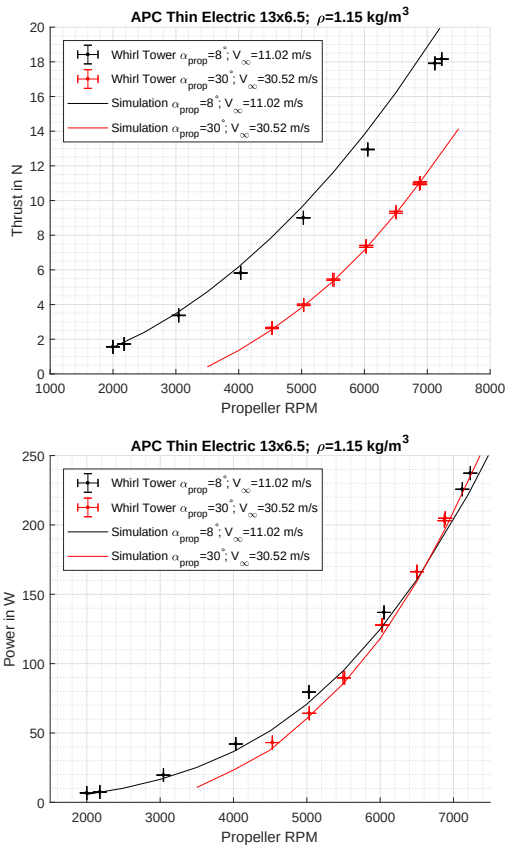


Figure 9: Agreement between measurements and simulations compared at $\alpha_{prop} = 8°$ and $\alpha_{prop} = 30°$.

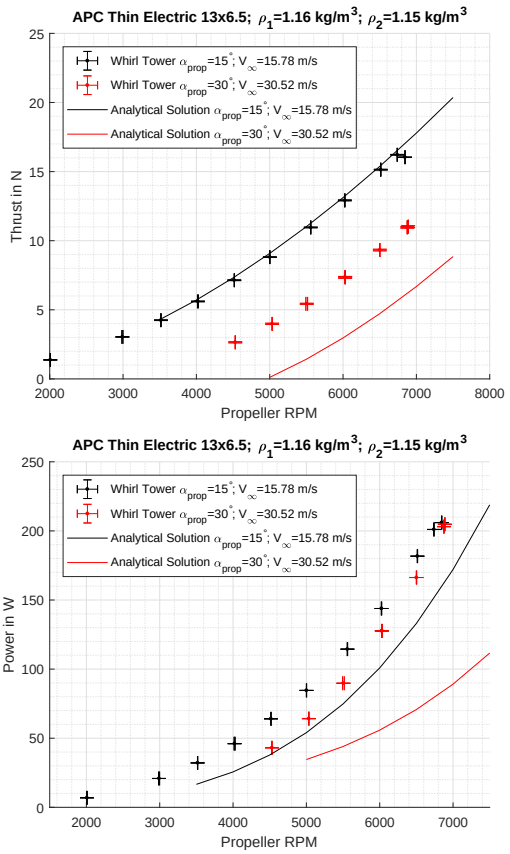


Figure 10: Analytical solution computed with Equation 8 and Equation 9.

of simulating the counter-rotating configuration is the simulation of the upper propeller as an isolated propeller and the simulation of the bottom propeller with a uniform inflow between v_i and $2v_i$ of the upper propeller. The highest uncertainty of this method is the estimation of the upper propellers wake decay. Some investigations on the wake of propellers from the authors can be found in [14].

The most interesting question that arises during the development of ANDroMeDA is how the influence of the top propeller on the bottom propeller changes when the unit is operated at higher angles of attack and higher inflow speeds. Due to strong vibrations at the test rig at forward flight conditions the full throttle test could not be repeated in the wind tunnel and another test methodology was applied: The bottom propeller was kept at a nominal thrust of about $T_{bottom} = 650\ g$ while the top propeller's thrust was slowly increased up to $T_{top} = 1000\ g$. This test gave some first insight views on the behavior in forward flight conditions compared to the static test case scenario. Table 2 compares the static test case to the forward flight case. It can be seen that in the forward flight case the bottom propeller suffers much less from the presence of the top propeller.

It is assumed that this fact is originated in the "cleaner" inflow condition for the bottom propeller in the forward flight case (Figure 11). Of course this hypothesis has to be confirmed by further measurements.

9 Conclusion and Outlook

It could be proven that static propeller measurements alone combined with blade element simulations can be sufficient to calculate the performance of a propeller in forward flight conditions as they occur at a fast flying multicopter. The error in the investigated operating conditions was always less than 10% and in the most cases a lot smaller. It has been observed that at the design operating point of the planned vehicle, ANDroMeDA even a slightly higher thrust can be achieved at nearly the same power than at static propeller operation. This could also be explained by simple momentum theory. As expected the thrust decreased greatly at higher angles of attack and higher speeds.

Another, simpler approach to model the performance of inclined propellers with less computational cost and a lot less programming effort proved to be only sufficient to predict the thrust at moderate angles of attack. At higher angles of at-

	ESC Signal	Motor Current	RPM
Static Case	+9 %	+75 %	+25 %
$\alpha_{prop} = 15°$; $V_\infty = 11.88m/s$	+4 %	+20 %	+6 %

Table 2: Changes necessary to keep the thrust of the bottom propeller steady at $T_{bottom} = 650\ g$ with the top propeller running at $T_{top} = 1000\ g$.

tack the thrust was underpredicted while the power with was always underpredicted this method. For future projects this semi-analytical approach could be adjusted to simulation or measurement data. But it seems unlikely that it is possible to model the propeller performance of arbitrary multicopter propellers at fast forwards speeds by this method without BE simulations or measurements.
Tests with a counter-rotating propeller arrangement revealed a positive effect of the propellers angle of attack and inflow speed on the operation of the bottom propeller. While the bottom propeller is influenced strongly by the wake of the top propeller this deficiency decreases at an inclined operation with some inflow speed. Unfortunately the test rig for counter-rotating propellers suffered from severe vibration so that only a few clean measurements could be taken. The proposed hypothesis has to be confirmed by further measurements in the future.

10 Acknowledgements

The ANWIND project was founded by the German Federal Ministry for Economic Affairs and Energy (BMWi). This support is gratefully acknowledged. The authors also would like to thank the Institute of Aerodynamics and Gas Dynamics at the University of Stuttgart making the wind tunnel tests possible. We would like to thank in particular Mr. Bernd Peters for his great support and advice.

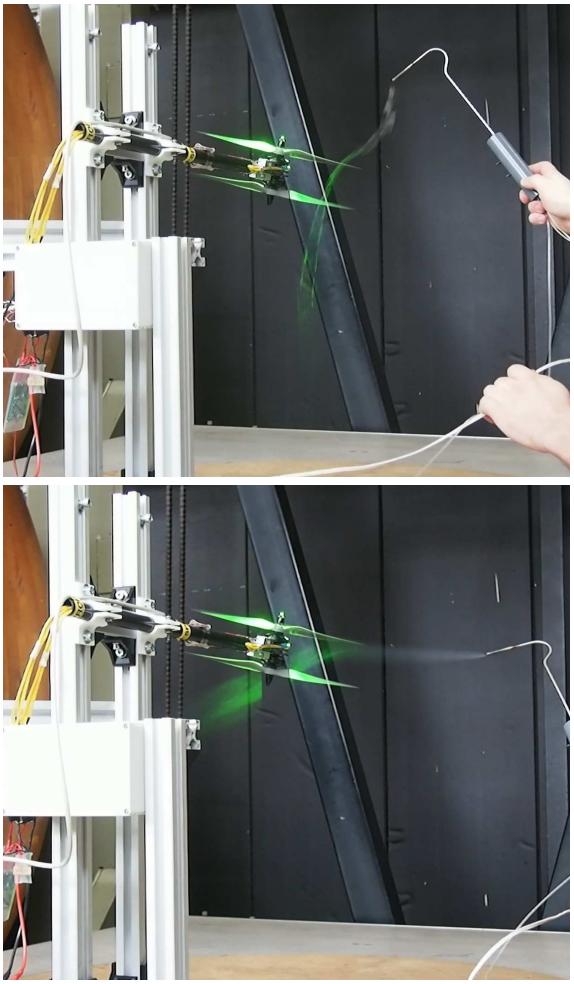


Figure 11: Smoke visualization of the counter-rotating propeller arrangement at $T = 12.5N$. Top: Static operation with wind tunnel turned off. Bottom: $\alpha_{prop} = 15°$ and $V_\infty = 11m/s$.

References

[1] M. Drela. Qprop formulation. *MIT Aero & Astro*, 2006.

[2] M. Hepperle. Javaprop user manual, 2008.

[3] M. Silvestre, J. Morgado, and J. Pascoa. Jblade: a propeller design and analysis code. In *2013 International Powered Lift Conference - AIAA AVIATION Forum*, 2013.

[4] M. Selig and J. Brandt. Propeller performance data at low reynolds numbers. *49th AIAA Aerospace Sciences Meeting AIAA 2011-1255*, 2011.

[5] W. Johnson. *A comprehensive analytical Model of Rotorcraft Aerodynamics and Dynamics*. Johnson Aeronautics, Palo Alto, California, 1988.

[6] Phidgets. Micro load cell (0-20kg) - czl635, 2016.

[7] AVIA Semiconductors. *24-Bit Analog-to-Digital Converter(ADC) for Weigh Scales*, 1 2015.

[8] W. Johnson. *Helicopter Theory*. Dover Publications, New York, NY, 2000.

[9] J. G. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge University Press, Cambridge, 2002.

[10] H. Glauert. *A general Theory of the Autogyro*. Presented by the Director of scientific Research Air Ministry, Reports and Memoranda No. 111, 1926.

[11] M. Drela and H. Youngren. Xfoil 6.9 user primer. *MIT Aero & Astro*, 2001.

[12] F. Bohorquez, F. Rankinsy, J.D. Baederz, and D.J. Pinesx. Hover performance of rotor blades at low

reynolds numbers for rotary wing micro air vehicles. an experimental and cfd study. In *21st Applied Aerodynamics Conference*, 2003.

[13] J. G. Leishman and A. Shreyas. An optimum coaxial rotor system for axial flight. *Journal of the American Helicopter Society*, 53(4):366–381, 2008.

[14] C. Molter and P.W. Cheng. to appear: Optimal placement of an airflow probe at a multirotor uav for airborne wind measurements. In *Proceedings of the 43rd European Rotorcraft Forum*, 2017.