



IMAV 2014

International Micro Air Vehicle Conference and Competition 2014

*August 12-15, 2014
Delft, The Netherlands*



Preface

It is our pleasure to present the proceedings of the International Micro Air Vehicle Conference and Competition, which was held in Delft, the Netherlands from August 12 to 15, 2014. This event, organized by Delft University of Technology, aims at the exchange of the most recent findings in the rapidly expanding area of Micro Air Vehicles (MAVs). The received submissions show that this is an active field of research all over the world.

These proceedings contain the peer-reviewed scientific papers and the technical papers presented at the IMAV 2014. The topics of these papers range from the design of human-friendly UAVs to a novel pitch and roll mechanism for controlling a flapping wing MAV. Together, the papers give an overview of the current state-of-the-art of the field of Micro Air Vehicles, indicating in which direction this field may develop.

We would like to thank the 20 members of the program committee, who reviewed all submissions and selected 38 papers, consisting of 11 technical papers and 27 scientific papers. 25 papers were selected for oral presentation at the conference.

Concerning the competition part of the IMAV 2014, it is the first time that the outdoor and indoor competition are held simultaneously, at an exercise village called 'Oostdorp'. A disaster scenario is simulated, in which emergency services need to get an overview of the situation as quickly as possible. To this end, the competition teams need to provide the emergency services with an up-to-date map of the disaster area, indicating which roads are still accessible and which ones are blocked. In addition, MAVs have to fly into the different houses to search for survivors of the disaster. All mission elements aim at further advancing the (autonomous) capabilities of MAVs, in the hope to bridge the gap between scientific research and application in real-world situations.

Finally, we want to express the hope that the specific nature of the IMAV event (a combination of a scientific conference and a real-world competition) in combination with the open access nature of the publications, will continue to advance the state-of-the-art in this relatively young research area.

July 2014

IMAV 2014 Program Chairs:

G.C.H.E. de Croon
Delft University of Technology

E. van Kampen
Delft University of Technology

C. De Wagter
Delft University of Technology

C.C. de Visser
Delft University of Technology

Program Committee

Dr. M. Amelink	THALES	Prof. S. Shkarayev	UA
Dr. D. Choukroun	DUT	Prof. W. Shyy	HKUST
Ir. C. De Wagter	DUT	Dr. V. Trianni	ISTC-CNR
Dr. G. Hattenberger	ENAC	Dr. A. Visser	UVA
Dr. N. Michael	CMU	Prof. P. Vörsman	TUB
Prof. J.M. Moschetta	ISAE	Dr. G. de Croon	DUT
Prof. J.A. Mulder	DUT	Dr. C.C. de Visser	DUT
Dr. M. Ol	USAF	Dr. E. van Kampen	DUT
Ir. B. Remes	DUT	Dr. B. van Oudheusden	DUT
Ir. R. Ruijsink	DUT	Dr. L. van der Maaten	DUT

Sponsored by

AerialTronics <http://www.aerialtronics.com/>
 Parrot <http://www.parrot.com/>
 The MathWorks <http://www.mathworks.nl/academia/student-competitions/imav/>
 One Bit Squared <http://1bitsquared.com/>
 Dutch Police <http://www.politie.nl/>
 Thales <https://www.thalesgroup.com/>



Supported by

TU Delft <http://mavlab.lr.tudelft.nl/>
 US Army International Technology Center-Atlantic <http://www.rdecom.army.mil/itcatlantic/>
 Office of Naval Research <http://www.onr.navy.mil/>
 Koninklijke Landmacht <http://www.defensie.nl/organisatie/landmacht>



Koninklijke Landmacht



Scientific Papers

[1] Dynamic Modeling of a Rotating Insect Wing	p.10
[2] Design, Fabr., and Perf. Test of Hovering-Based Flapping-Wing MAV Capable of Sustained & Controlled Flight ..	p.18
[3] Twirre: Architecture for autonomous mini-UAVs using interchangeable commodity components	p.26
[4] Pitch and Yaw Control of Tailless Flapping Wing MAVs by Implementing Wing Root Angle Deflection	p.34
[5] Design Analysis, Modelling and Experimental Validation of a Bird-like Flapping-Wing Flying Robot	p.42
[6] Off-board Visual Odometry and Control of an Ultralight Quadrotor MAV	p.50
[7] Actuator Fault Detection and Diagnosis for Quadrotors	p.58
[8] eMotionSpheres: A New Design for Human-Friendly UAVs	p.64
[9] Stereo vision with consumer grade high resolution cameras for a micro air vehicle	p.69
[10] Fullscale Windtunnel Invest. of Actuator Effectiv. during Stat. Flight within Entire Flight Envelope of Tiltwing MAV	p.77
[11] Robust marker-tracking system for vision-based autonomous landing of VTOL UAVs	p.84
[12] Study of design parameters of flapping-wings	p.92
[13] Using a gyroscope to stabilize the attitude of a fly-sized hovering robot	p.102
[14] Effect of chordwise deformation on unsteady aerodynamic mechanisms in hovering flapping flight	p.110
[15] Pitch and roll control mechanism for a hovering flapping wing MAV	p.118
[16] A Computational Study on the Aerodynamics of a 90 mm Ducted Contra-Rotating Lift Fan	p.126
[17] Towards Estimation and Correction of Wind Effects on a Quadrotor UAV	p.134
[18] Obstacle avoidance by combining background subtraction, optical flow and proximity estimation	p.142
[19] Trajectory-aware Ad hoc Routing Protocol for Micro Aerial Vehicle Networks	p.150
[20] Development of the flight model of a tilt-body MAV	p.157
[21] Deformation Analysis of a Hummingbird Inspired MAV Flapping Wing using Digital Image Correlation	p.164
[22] V-REP & ROS Testbed for Design, Test & Tuning of QR Vision Based Fuzzy Ctrl Syst. for Auton. Landing ...	p.172
[23] Development and Application of PP-CNT Composite for Hummingbird Inspired MAV Flapping Wings	p.180
[24] An Ecological Approach to the Supervisory Control of UAV Swarms	p.188
[25] Modelling of a Hybrid UAV Using Test Flight Data	p.196
[26] Vortex Formation and Force Generation Mechanisms of the DelFly II in Hovering Flight	p.204
[27] Determination of trim curves for a flapping-wing MAV	p.212

Technical Papers

[28] Design and development of UGS flapping wing MAVs	p.219
[29] The Next Generation Design of Autonomous MAV Flight Control System SmartAP	p.225
[30] Automatic extraction of moving obj. from UAV-borne monocular images using multi-view geometric constraints	p.230
[31] Design of Ducted Fans of Small Height for Hexacopter with Long Hover Ability	p.235
[32] Design and Build of Swarm Quadrotor UAVs at UGS	p.238
[33] Using the Paparazzi UAV System for Scientific Research	p.247
[34] Autonomous Aerial Mapping and Observation Task for IMAV2014 Competition	p.253
[35] Flexible framework for the development of versatile MAV systems for multi-disciplinary applications	p.259
[36] Design of a Quadrotor System for an Autonomous Indoor Exploration	p.265
[37] Modular Quad Rotor (Semi-Autonomous/Autonomous)	p.272
[38] Lisa-S 2.8g autopilot for GPS-based flight of MAVs	p.280

Authors

A

Ageev, Nikita p.235
 Alonso-Mora, Javier p.134
 Arkhipov, Maxim p.235
 Armanini, Sophie p.212

B

Beardsley, Paul p.134
 Bestmann, Ulf p.265
 Bijl, Hester p.110
 Borst, Clark p.188
 Bronz, Murat p.247
 Bruin, Dick p.26

C

Chan, Woei Leong p.34, p.18
 Chand, Aneesh p.42
 Chen, Ben M. p.50
 Chen, Weixiong p.126
 Chirarattananon, Pakpong p.102
 Chu, Qiping p.196
 Colles, Emmanuel p.253
 Cordero, Miguel p.259
 Croon, Guido de p.188, p.196, p.212, p.280

D

Debiasi, Marco p.34, p.18
 Defay, François p.157
 Deng, Shuanghou p.204
 Dijkstra, Klaas p.26
 Díaz, Luis p.259

E

Esdén-Tempski, Piotr p.280

F

Foo, Qiwan p.238
 Fuchs, Christian p.188
 Fuller, Sawyer p.102

G

Gaissert, Nina p.64
 Goosen, Johannes F.L. p.92

Gorraz, Michel p.247
 Goyal, Tigmanshu p.164, p.180
 Gäbel, Mario p.265

H

Hartmann, Philipp p.77
 Hattenberger, Gautier p.247, p.253
 Helbling, Elizabeth p.102
 Horst, Erik van der p.196
 Hua, Alexandre p.118
 Huang, Rui p.50

J

Jankauski, Mark p.10
 Jebens, Agalya p.64
 Jebens, Kristof p.64
 Jiménez, Antonio p.259
 Jumat, Muhammad Ridhwan p.219
 Jumat, Ridhwan p.238

K

Kamle, Sudhir p.164, p.180
 Kampen, Erik-Jan van p.58
 Kannan, Sumasundar p.172
 Karasek, Matej p.118
 Kawanishi, Michihiro p.42
 Keulen, Fred Van p.92
 Keuning, Wouter p.26
 Kimura, Motoki p.230
 Knubben, Elias Maria p.64
 Koh, Wei Chong p.238
 Krüger, Thomas p.265
 Kumar, David p.164, p.180
 Kumar, Vemuri Shyam p.164, p.180

L

Lai, Shupeng p.50
 Lalami, Mohamed Esseghir p.118
 Lauri, Mikko p.69
 Lee, Tong Heng p.50

Li, Kun	p.50
Li, Ruiyu	p.126
Loh, Samuel	p.272
Loosdrecht, Jaap van de	p.26
Lu, Peng	p.58
Lustosa, Leandro R.	p.157

M

Melin, Joonas	p.69
Mohite, P. M.	p.164, p.180
Moormann, Dieter	p.77
Moschetta, Jean-Marc	p.157
Mugrauer, Günter	p.64
Mugrauer, Rainer	p.64
Mulder, Max	p.188
Muzaffar, Raheeb	p.150

N

Nagai, Masahiko	p.230
Nan, Yanghai	p.118
Narikiyo, Tatsuo	p.42
Ng, Alvin	p.238
Nguyen, Quoc-Viet	p.34, p.18
Nollet, Fabien	p.253
Noyon, Tijs	p.110

O

Olivares-Mendez, Miguel A.	p.172
Oudheusden, Bas Van	p.110
Oudheusden, Bas W. Van	p.204

P

Paassen, René van	p.188
Percin, Mustafa	p.204
Phang, Swee King	p.50
Postma, Jorn Haaije	p.26
Preumont, André	p.118

R

Rabe, Clemens	p.64
Rayar, Tony Victor	p.238
Remes, Bart	p.196, p.204, p.280
Ritala, Risto	p.69
Rudin, Konrad	p.134
Ruíz, Jonathan	p.259

S

Sagar, Jaysinh	p.142
Schiano, Fabrizio	p.134
Schutter, Joris De	p.84
Schütt, Marten	p.77
Serokhvostov, Sergey	p.235
Shao, Xiaowei	p.230
Sharma, Akash	p.164
Shen, I.Y.	p.10
Shibasaki, Ryosuke	p.230
Shilov, Kirill	p.225
Siciliano, Bruno	p.134
Sieglwart, Roland	p.134
Smeur, Ewoud	p.196, p.280
Srigrarom, Sutthiphong	p.219, p.238

T

Tan, Benson	p.272
Tan, Ping	p.50
Tay, Wee Beng	p.110
Tenaglia, Andrea	p.204
Theys, Bart	p.84
Tienen, Freek van	p.280
Trujillo, Miguel Ángel	p.259

V

Vandeportaele, Bertrand	p.253
Vaswani, Vishal	p.272
Verbandt, Maarten	p.84
Verboom, Jochem	p.212
Viguria, Antidio	p.259
Visser, Arnoud	p.142
Visser, Coen de	p.212
Voos, Holger	p.172

W

Wagter, Christophe De	p.196, p.280
Wang, Fei	p.50
Wang, Haitong	p.126
Wang, Qi	p.92
Wang, Yangang	p.126
Wood, Robert	p.102

Y

Yanmaz, Evsen	p.150
Yu, Bin	p.58

Conference Oral Presentations

Session 1: Autonomous flight using vision

Robust marker-tracking system for vision-based autonomous landing of VTOL UAVs	p.84
Obstacle avoidance by combining background subtraction, optical flow and proximity estimation	p.142
V-REP & ROS Testbed for Design, Test & Tuning of QR Vision Based Fuzzy Ctrl Syst. for Auton. Landing	p.172

Session 2: Flapping wing design

Design, Fabr., and Perf. Test of Hovering-Based Flapping-Wing MAV Capable of Sustained & Controlled Flight	p.18
Design Analysis, Modelling and Experimental Validation of a Bird-like Flapping-Wing Flying Robot	p.42
Study of design parameters of flapping-wings	p.92

Session 3: Flapping wing control

Pitch and Yaw Control of Tailless Flapping Wing MAVs by Implementing Wing Root Angle Deflection	p.34
Using a gyroscope to stabilize the attitude of a fly-sized hovering robot	p.102
Pitch and roll control mechanism for a hovering flapping wing MAV	p.118

Technical session

Automatic extraction of moving obj. from UAV-borne monocular images using multi-view geometric constraints ...	p.230
Using the Paparazzi UAV System for Scientific Research	p.247
Flexible framework for the development of versatile MAV systems for multi-disciplinary applications	p.259
Design of a Quadrotor System for an Autonomous Indoor Exploration	p.265
Lisa-S 2.8g autopilot for GPS-based flight of MAVs	p.280

Session 4: Flight mechanics

Fullscale Windtunnel Invest. of Actuator Effectiv. during Stat. Flight within Entire Flight Envelope of Tiltwing MAV	p.77
Towards Estimation and Correction of Wind Effects on a Quadrotor UAV	p.134
Development of the flight model of a tilt-body MAV	p.157

Session 5: Flapping wing aerodynamics

Effect of chordwise deformation on unsteady aerodynamic mechanisms in hovering flapping flight	p.110
Vortex Formation and Force Generation Mechanisms of the DelFly II in Hovering Flight	p.204

Session 6: Modelling

Dynamic Modeling of a Rotating Insect Wing	p.10
Modelling of a Hybrid UAV Using Test Flight Data	p.196
Determination of trim curves for a flapping-wing MAV	p.212

Session 7

Actuator Fault Detection and Diagnosis for Quadrotors	p.58
eMotionSpheres: A New Design for Human-Friendly UAVs	p.64
An Ecological Approach to the Supervisory Control of UAV Swarms	p.188

Release

V2.0, Rev.16910, 2026-10-24 09:33

V1.0: August 2014. V2.0: rendered in the 2018-2026 format, with the same page numbers.

Created by: G. de Croon, E.J. van Kampen, C. De Wagter and C. de Visser

Chapter 1

Scientific Articles

Dynamic Modeling of a Rotating Insect Wing

Mark Jankauski*, I.Y. Shen

University of Washington, 371 Loew Hall, Seattle, WA 98195

ABSTRACT

A dynamic model of an insect wing is developed treating the wing as a deformable body subject to three-dimensional finite rotation about a fixed point at the base of the wing. Discretization of a stationary wing is conducted via finite element analysis to determine the natural frequencies and mode shapes of the wing. By formulating and discretizing the kinetic and potential energy, we derive the equation of motion governing the modal response of a flapping wing using Lagrange's equation. The equation of motion indicates Coriolis, Euler, and centrifugal forces resulting from the finite rotation are responsible for the wings elastic deformation. Numerical integration reveals a beat phenomenon that arises from the Coriolis excitation in the first vibration mode. The beat phenomenon is insensitive to yaw amplitudes and non-zero initial conditions but diminishes in the presence of damping. The beat phenomenon can potentially be used to estimate gyroscopic forces.

1 INTRODUCTION

Insect flight has recently been a subject of significant research within biology and engineering communities. For biologists, such studies provide critical insights into the sensorimotor coordination of movement in animals. For example, knowledge of the instantaneous wing shape helps determine the direction and magnitude of aerodynamic forces acting upon the wing. This leads to an understanding of lift-to-drag ratio, thrust production and aerodynamic efficiency of insect flight [1, 2]. Moreover, wing deformation activates strain receptors (mechanosensation cells), which provides the insect brain with feedback used to identify flight status. Insect wings are richly imbued with strain sensors [3].

For engineers, insect flight research can help realize autonomous flying objects, such as flapping-wing micro-aerial vehicles (MAVs). Flapping-wing MAVs are capable of performing difficult maneuvers and stationary hovering while consuming relatively little power [4]. These characteristics are desirable for indoor flight, and consequently have generated interest in potential military surveillance and reconnaissance applications [5]. While several MAVs have been successfully developed, there remains a necessity for design

optimization in order to reduce weight, power consumption, and cost [6]. A functional dynamic wing model may facilitate this effort.

Several models have previously been developed to describe the complex dynamics of an insect wing during flight. Many kinematic and dynamic models treat the insect body and wings as several connected rigid bodies. For example, a kinematic model developed by Rocca et al. treats the wing as a rigid body subject to three-dimensional rotation about a fixed point on the insect body [7]. An alternative dynamic model developed by Orlowski and Girard also treats the insect forewing as a rigid body, and considers aerodynamic loading as a generalized force acting on the wing [8]. Such kinematic and dynamic models can be used to effectively estimate wing velocity and aerodynamic force production while retaining a lower number of degrees of freedom. However, by treating the wing as a rigid body, such models are unable to provide information about localized strains, and therefore have limitations in determining insect neural feedback or designing strain-based control systems for MAVs.

Alternatively, researchers have developed finite element analysis (FEA) based models [9]. Finite element models are capable of estimating localized strains on a deformed wing for each mode shape. However, FEA models are limited to infinitesimal rotation and cannot describe the entire time-dependent strain response of a flapping wing. The large rigid body motion and rotation of the wing generate a time-varying stiffness matrix $\mathbf{K}(t)$. The implication of this is that the wing will have to be re-meshed during each time step of the analysis, resulting in a larger number of degrees of freedom. The significant number of degrees of freedom makes a direct FEA model computationally impractical. Physical insights, such as the contribution of gyroscopic forces, may also be lost.

Each of these models is capable of independently describing aspects of insect flight, such as aerodynamic force production and strain. However, there remains a necessity to unify these models such that all their positive attributes can be fully realized. By synthesizing characteristics from each of these models, we develop in this paper a reduced-order model of a flapping insect wing with the following important characteristics. First, the model is capable of predicting strain components under finite rotation. Second, the model retains a low order to maintain computational feasibility, thereby facilitating parametric studies. Third, the model can accommodate nonspecific geometry and therefore can be used to describe any flapping insect wing. Lastly, the model discretizes the wing via finite element analysis, so arbitrary geometry can be implemented readily. As a first attempt at model formulation,

*Email address: mjankaus@uw.edu

effects of aerodynamics and aeroelasticity are neglected. It is assumed elastic deformation of the wing is solely a result of inertial-elastic effects [2].

The remainder of the paper is organized as follows. First, the dynamic model is formulated using Lagrange mechanics. The model is then applied to a *Manduca sexta* forewing, revealing a beat phenomenon in the first two modes caused by the interaction of gyroscopic and centrifugal forces with the free vibration of the wing. Various factors that may affect the presence of the beat phenomenon are then investigated, including initial conditions, yaw rotation amplitude, and damping.

2 FORMULATION

The model assumes the wing as a deformable body subject to three-dimensional rotation about a point fixed on the insect's body. The displacement of any point on the wing is described by two finite rotations (pitch and roll), one small rotation (yaw), and one infinitesimal out-of-plane deflection. The rigid body displacement of the wing is a kinematic process, and can readily be determined by establishing a rotating coordinate frame. The out-of-plane deflection is characterized as a dynamic process, and must be calculated by solving the equations of motion derived through potential and kinetic energy terms.

2.1 Kinematics

The wing is placed into a fixed reference frame, with the origin representing the fixed point of rotation and I, J, K being three orthogonal unit vectors associated with the fixed reference frame. (Fig 1). For the purposes of this formulation, I, J, K will also be referred to as coordinate axes. A rotating reference frame that pitches and rolls but does not yaw with the wing is established. The first rotation (roll) has an amplitude α and occurs about the I -axis, transforming the reference frame to the $I' - J' - K'$ coordinate system. The second rotation (pitch) has an amplitude of β and occurs about the J' -axis, transforming the reference frame to the $I'' - J'' - K''$ coordinate system. Both of these rotations are assumed to have large and finite amplitudes. The third rotation corresponds to the yaw rotation, and is denoted by γ . The yaw rotation (Figure 2) is assumed small, and therefore small angle approximations are employed. Consequently, the yaw rotation is not considered in the rotating reference frame. Through simple geometric analysis, the angular velocity $\vec{\Omega}$ of the rotating coordinate frame $I'' - J'' - K''$ is:

$$\vec{\Omega} = \dot{\alpha} \cos \beta I'' + \dot{\beta} J'' + \dot{\alpha} \sin \beta K'' \quad (1)$$

With the rotating coordinate frame established, a position vector $\vec{R}_n$ (Fig 2) is drawn in the $I'' - J'' - K''$ reference frame from the point of rotation to a differential mass dm on the wing geometry. $\vec{R}_n$ is the sum of three intermediate position vectors, $\vec{r}_1, \vec{r}_2$, and $\vec{r}_3$ (i.e., $\vec{R}_n = \vec{r}_1 + \vec{r}_2 + \vec{r}_3$), where $\vec{r}_1$ represents the position of the differential mass without yaw,

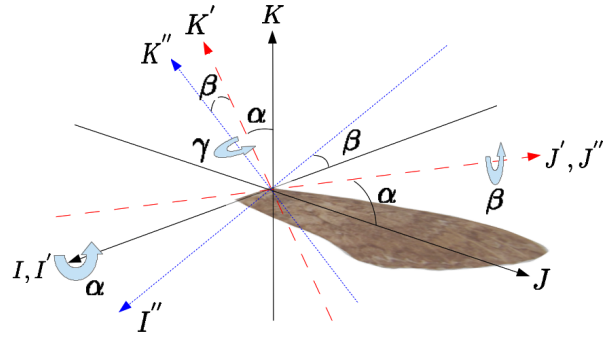


Figure 1: Development of a rotating reference frame

$\vec{r}_2$ represents the contribution of the yaw rotation to the position, and $\vec{r}_3$ represents the out-of-plane elastic deformation of the wing. In-plane deformation is assumed to be small, and is neglected in this formulation. Specifically, $\vec{r}_1, \vec{r}_2$, and $\vec{r}_3$ are described in both Cartesian and cylindrical coordinates as follows:

$$\vec{r}_1 = xI'' + yJ'' = r\vec{e}_r \quad (2)$$

$$\vec{r}_2 = \gamma(-yI'' + xJ'') = \gamma r\vec{e}_\theta \quad (3)$$

$$\vec{r}_3 = W(\vec{r}_1, t)K'' = W(\vec{r}_1, t)\vec{e}_z \quad (4)$$

Above, x and y are the coordinates of the differential mass and $W(\vec{r}_1, t)$ is the elastic deformation dependent on both time and position. The position vector $\vec{R}_n$ can then be differentiated with respect to time to determine the velocity of the differential mass. As dm is in a rotating coordinate frame, the chain rule must be applied to account for the time derivative of the coordinate frame. The resulting expression for the differential mass velocity $\dot{\vec{R}}_n$ is:

$$\dot{\vec{R}}_n = \vec{\Omega} \times \vec{R}_n + \dot{\vec{r}}_2 + \dot{\vec{r}}_3 \quad (5)$$

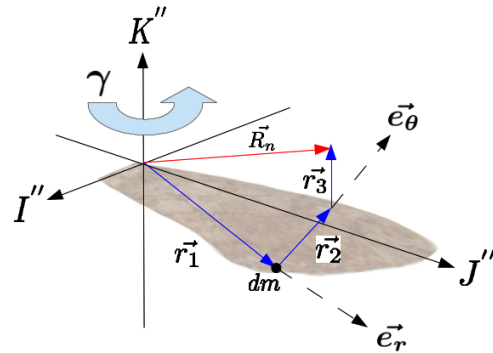


Figure 2: Position vectors and yaw rotation

2.2 Kinetic and Potential Energies

The kinetic energy T of the wing is

$$T = \frac{1}{2} \int_m \dot{\vec{R}}_n \cdot \dot{\vec{R}}_n dm \quad (6)$$

where Equation 6 is integrated over the domain of the wing. The potential energy V is

$$V = \frac{1}{2} \int_v S(W, W) dv \quad (7)$$

where $S(W, W)$ represents a quadratic and symmetric strain energy density function, i.e., $S(a, b) = S(b, a)$, and the domain of integration is the volume of the wing. The use of the strain energy density function allows this formulation to be applied for any elastic body with complex geometry.

2.3 Discretization

The unknown deformation of the wing $W(\vec{r}_1, t)$ can be expressed via an eigenfunction expansion

$$W(\vec{r}_1, t) = \sum_{k=1}^{\infty} \phi_k(\vec{r}_1) q_k(t) \quad (8)$$

where $\phi_k(\vec{r}_1)$ is the k^{th} mode shape and $q_k(t)$ is its modal response to be determined. Moreover, the mode shapes $\phi_k(\vec{r}_1)$ are normalized with respect to the mass satisfying the following orthonormal conditions,

$$\int_m \phi_k \phi_e dm = \delta_{ke} \quad (9)$$

where δ_{ke} is the Kronecker delta. Similarly, $\phi_k(\vec{r}_1)$ experiences orthogonality via the strain energy density function,

$$\int_v S(\phi_k, \phi_e) dv = \omega_k^2 \delta_{ke} \quad (10)$$

where ω_k is the natural frequency of the k^{th} mode shape. The advantage to the eigenfunction expansion is that the mode shape $\phi_k(\vec{r}_1)$ and the natural frequency ω_k may be determined using finite element modeling. Therefore, this formulation is valid for any arbitrary wing shape. Moreover, the boundary conditions of the wing are accounted for directly in the finite element analysis, and need not be evaluated explicitly for development of the equations of motion.

Two geometric vectors useful to the derivation of the equations of motion are defined as follows

$$\vec{a}_k = \int r \phi_k(\vec{r}_1) \vec{e}_r dm = \int \phi_k (xI'' + yJ'') dm \quad (11)$$

$$\vec{b}_k = \int r \phi_k(\vec{r}_1) \vec{e}_\theta dm = \int \phi_k (-yI'' + xJ'') dm \quad (12)$$

where $\vec{a}_k$ represents a geometric vector corresponding to the position of the inertial force center of the k^{th} mode shape relative to the point of rotation, and $\vec{b}_k$ (Equation 12) represents a 90° counter-clockwise rotation from $\vec{a}_k$.

2.4 Equation of Motion

To develop the equations of motion, kinetic and potential energy terms are discretized via Equation 8 and then subjected to Lagrange's equation. The modal response of the k^{th} mode, $q_k(t)$, is the generalized coordinate. The resulting equation of motion is determined as

$$\ddot{q}_k + [w_k^2 - (\dot{\alpha}^2 \cos^2 \beta + \dot{\beta}^2)] q_k = -\dot{\vec{\Omega}} \cdot (\gamma \vec{a}_k - \vec{b}_k) - (K'' \cdot \vec{\Omega}) [\vec{\Omega} \cdot (\vec{a}_k + \gamma \vec{b}_k)] - 2\dot{\gamma} \vec{\Omega} \cdot \vec{a}_k \quad (13)$$

Equation 13 is a linear, time-varying second order ordinary differential equation. Time-variance is introduced in the stiffness coefficient, $[w_k^2 - (\dot{\alpha}^2 \cos^2 \beta + \dot{\beta}^2)]$, and is a direct result of centrifugal softening. Forcing terms appear on the right hand side of Equation 13 as a result of the wing's rotation. Interestingly, three forces typical in rotating coordinate frames are identified: the Coriolis force, Euler force, and centrifugal force. Each of these forces has a corresponding vector projection that is described in terms of $\vec{a}_k$ and $\vec{b}_k$. Table 1 shows the vector projection of each forcing term specific to the wing.

Forcing Term	Projected Representation
Euler Force	$-\dot{\vec{\Omega}} \cdot (\gamma \vec{a}_k - \vec{b}_k)$
Centrifugal Force	$-(K'' \cdot \vec{\Omega}) [\vec{\Omega} \cdot (\vec{a}_k + \gamma \vec{b}_k)]$
Coriolis Force	$-2\dot{\gamma} \vec{\Omega} \cdot \vec{a}_k$

Table 1: Summary of Excitation Terms

Due to the linearity of Equation 13, solutions for $q_k(t)$ may be determined independently for each forcing term outlined in Table 1. This result is meaningful, as it allows direct comparison between the generalized coordinate response $q_k(t)$ for each type of force.

3 EXAMPLE

To demonstrate the usage of Equation 13, the proposed model is applied to the forewing of the hawkmoth *Manduca sexta*. The *Manduca sexta* is an ideal subject for insect flight study due to the small variation between individual specimens and the abundance of available research [10]. To apply Equation 13, a finite element model is constructed to determine the mode shapes and natural frequencies of a *Manduca sexta* forewing. Next, assumed rotation profiles are presented for the insect in forward flight. The excitation terms and generalized coordinate responses for the first two mode shapes are then analyzed. Finally, the effects of initial conditions, yaw

rotation amplitudes, and damping on generalized coordinate responses are explored.

3.1 Finite Element Modeling

Finite element modeling is used to find mode shapes and natural frequencies of the wing. Published research has determined the natural frequencies and mode shapes of the *Manduca sexta* forewing via finite element analysis, and the results agree closely with experimental results [11]. These finite element models serve as a benchmark for the finite element model developed for this research.

A to-scale finite element model of the wing is constructed in ABAQUS by tracing computed tomography (CT) images of a *Manduca sexta* forewing. A camber of approximately 13% of the wing chord length is implemented, and has the notable effect of providing support to the first bending mode. Due to the aspect ratio of the wing, shell elements are utilized. Venation patterns of a real *Manduca sexta* forewing are neglected, and paper material is used in lieu of the actual membrane material. The properties of the paper used in the finite element model are given in Table 2.

Symbol	Description	Value	Units
t	Thickness	81	μm
E	Young's Modulus	10	MPa
ρ	Density	0.86	g/cm^3
ν	Poisson's Ratio	0.3	-

Table 2: Assumed material properties for the finite element paper wing

The natural frequencies determined by the finite element model are shown in Table 3, and the corresponding mode shapes are shown in Figure 3. The natural frequencies of the first two mode shapes agree within 4.5% of the reported values, indicating that the paper wing can adequately describe the dynamics of an actual *Manduca sexta* forewing.

Natural Frequency	Determined Value	Reported Value
ω_1	62.5 Hz	60 Hz
ω_2	86.9 Hz	84 Hz

Table 3: Comparison of paper wing natural frequencies and reported *Manduca sexta* natural frequencies [11]

Vectors $\vec{a}_k$ and $\vec{b}_k$ are then determined numerically from the mode shapes pictured in Figure 3 using the following approximation.

$$\vec{a}_k = \sum_{i=1}^j \phi_{i,k}(x_i, y_i) [x_i I'' + y_i J''] dm_i \quad (14)$$

$$\vec{b}_k = \sum_{i=1}^j \phi_{i,k}(x_i, y_i) [-y_i I'' + x_i J''] dm_i \quad (15)$$

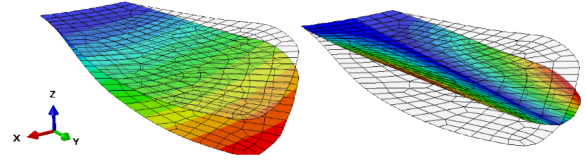


Figure 3: First two mode shapes of the paper wing- (Left) First Spanwise Bending, (Right) First Spanwise Torsion

Above, the i index denotes the i^{th} node of the finite element model and the j index represents the total number of nodes. Therefore, x_i and y_i represent the $x - y$ coordinates of the i^{th} node, $\phi_{i,k}(x_i, y_i)$ represents the k^{th} mode shape of the i^{th} node, and dm_i represents the nodal mass of the i^{th} node. Table 4 shows the geometric vectors $\vec{a}_k$ and $\vec{b}_k$ for the first two modes. In general, $\vec{a}_k$ and $\vec{b}_k$ are very small. Therefore, a second set of vectors $\vec{a}'_k$ and $\vec{b}'_k$ is defined, where $\vec{a}'_k$ and $\vec{b}'_k$ are normalized with respect to the largest component of vectors $\vec{a}_k$ and $\vec{b}_k$. For predictions of physical quantities (i.e., strain distribution), $\vec{a}_k$ and $\vec{b}_k$ will be used. For comparison of modal responses, $\vec{a}'_k$ and $\vec{b}'_k$ will be used due to the convenience of their magnitudes.

Vector	I''	J''
$\vec{a}_1$	0.7552×10^{-5}	0.0934×10^{-5}
$\vec{b}_1$	-0.0934×10^{-5}	0.7552×10^{-5}
$\vec{a}_2$	-0.1728×10^{-5}	-0.0033×10^{-5}
$\vec{b}_2$	0.0033×10^{-5}	-0.1728×10^{-5}

Table 4: Geometric Vectors for first two mode shapes- Paper Wing FEA model

3.2 Rotation Profiles

Equation 13 allows for any function to describe pitch, roll, and yaw rotations independently. In this example, each rotation is assumed sinusoidal and periodic with a frequency of ω_d representing the flap frequency of the *Manduca sexta*. Mathematically, the roll rotation takes the form $\alpha = \alpha_0 \sin(\omega_d t)$, where α_0 denotes the rotation amplitude in radians. The pitch and yaw rotations take the same form, with their amplitudes described by β_0 and γ_0 respectively. Phase angles are not considered in this example.

The flight of the *Manduca sexta* is divided into two principal regimes— forward flight and hovering flight. In each of these regimes, rotation amplitudes α_0 , β_0 and γ_0 exhibit vastly different characteristics. Typical rotation amplitudes for a *Manduca sexta* traveling forward at a speed of 5 m/s are shown in Table 5 [10]. For this example, only the forward flight regime is considered. The flapping frequency used is $\omega_d = 25$ Hz, which falls within the observed 24.8-26.5 Hz range of flap frequencies [10].

Rotation Parameter	Amplitude (rad)
α_0	0.8
β_0	0.5
γ_0	0.2

Table 5: Rotation Amplitudes for Forward Flight (5 m/s) [10]

3.3 Time/Frequency Domain of System Parameters

Each of the system parameters is investigated in time and frequency domains. The system parameters correspond to the time-varying stiffness and excitation terms of Equation 13. Figure 4 shows each of the system parameters as a function of time for the first mode shape.

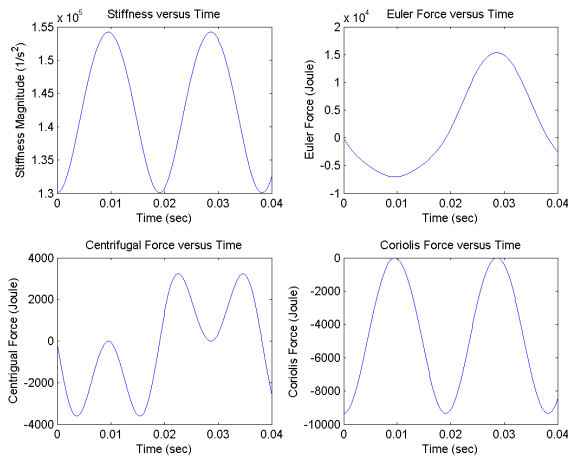


Figure 4: Systems Parameters in the time domain- First mode shape

From Figure 4, it can be seen that the Euler Force has the largest magnitude, the Coriolis Force has the second largest magnitude, and the centrifugal force has the smallest magnitude. Both the stiffness and Coriolis force vary nearly sinusoidally in sync with a frequency of approximately twice the flap frequency. The Euler and centrifugal forces appear to have more complicated forms. To determine the frequency content of each of these system parameters, a power spectral density (PSD) algorithm is employed.

The PSD decomposes each of the system parameters into its frequency components. The Coriolis Force and stiffness oscillate at a frequency close to $2\omega_d$. The Centrifugal force oscillates predominately at $3\omega_d$, with sub harmonics occurring at ω_d . The Euler force oscillates at ω_d , with super harmonics occurring at $2\omega_d$, $3\omega_d$, etc. In this case, super harmonics $3\omega_d$ and above tend to have a negligible effect. Oscillation frequencies of the Coriolis force occur near the first natural frequency ω_1 , and oscillation frequencies of the centrifugal force occur near the second natural frequency ω_2 . The prox-

imity to natural frequencies may have a marked effect on the generalized coordinate responses.

3.4 Generalized Coordinate Responses

Equation 13 is solved assuming zero initial conditions (e.g. $q_k(0) = 0, \dot{q}_k(0) = 0$). The response subject to each excitation term is solved individually, and due to linearity, the results are summed to determine the overall system response. In this section, all responses are solved numerically using Matlab. The results for the first mode are shown in Figure 5.

The generalized coordinate responses for the first mode provides interesting results. First, despite having the largest magnitude, the Euler force generates only the second largest response. The Euler force generates oscillations predominately at a frequency ω_d , roughly half the natural frequency of the first mode ω_1 . On the other hand, the Coriolis force oscillates at $2\omega_d$, which is close to ω_1 . As a result, the Coriolis force generates a larger response due to operating near resonance despite having a lesser magnitude than the Euler force.

Second, a clear beat phenomenon exists in the Coriolis response. The beat phenomenon occurs between the $2\omega_d$ oscillation frequency of the Coriolis force and effective natural frequency ω_1 of the first mode shape. Lastly, the centrifugal force appears to have a very minor contribution to the response of the first mode shape. This is due to the relatively low magnitude of the centrifugal force coupled with the fact that dominant oscillations occur away from the first natural frequency.

Next, the response of the second mode shape is solved. Similar to the Coriolis force for the first mode shape, the centrifugal force oscillates near the second natural frequency ω_2 . The result is a significant contribution to the generalized coordinate response, as well as the presence of a beat phenomenon occurring between $3\omega_d$ and the effective natural frequency of the second mode shape. Both the Euler force and Coriolis force oscillate away from resonance, and therefore have comparatively low contributions to the response.

4 BEAT PHENOMENON

Numeric simulations indicate the beat phenomenon occurs in the first and second modes as a response to Coriolis and centrifugal forcing, respectively. Such a response is a result of the forced vibration interacting with the free vibration of the wing. The beat phenomenon as revealed can also be used as a way to detect gyroscopic forces. The beat amplitude is proportional to the yaw amplitude γ_0 . It remains an open question as to how insects detect gyroscopic forces. The beat phenomenon, both in frequency and magnitude, may potentially allow insects to detect such gyroscopic forces.

Frye hypothesizes that the frequency content generated by the stretch receptor in the *Manduca sexta* forewing plays a vital role in the neural feedback used by the insect to interpret flight status [12]. It seems plausible then that the beat phenomenon has biological significance in providing neural

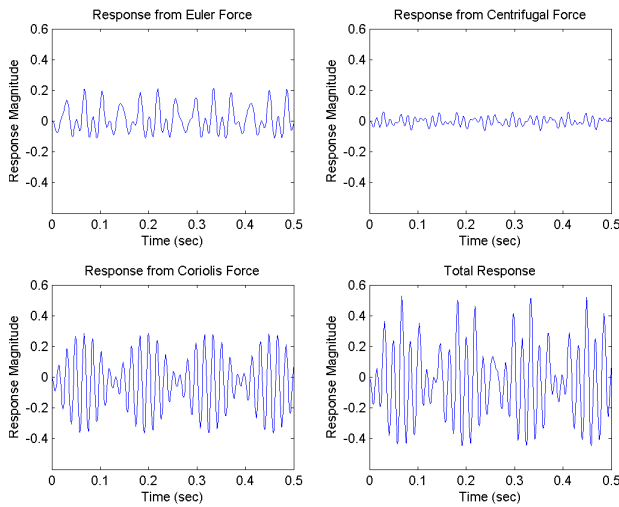


Figure 5: Generalized Coordinate Responses to Excitation Terms- First Mode

feedback during insect flight. As the total generalized coordinate response exhibits a beat response, the strain components in the wing will also reflect a beat response. Should the insect be capable of interpreting beat effects in the first two modes, the rotations α, β and γ may be identified. As these rotations have a direct correlation to aerodynamic parameters (e.g. lift, drag), the beat amplitudes may be used to identify flight status. Consequently, it is possible that MAV design could implement strain-based control systems rather than accelerometer-based control systems. This may potentially reduce both the weight and the power consumption of the MAV. Motivated by the potential applications of the beat phenomenon, we determine the sensitivity of the beat response in the first mode subject to Coriolis forcing.

4.1 Sensitivity to Yaw Amplitude

The amplitude of the yaw rotation, γ_0 , has been found to have a tremendous influence on the magnitude of the Coriolis force. While the typical yaw amplitudes for forward flight (Table 5) are relatively consistent, individual specimens may deviate slightly from these values. Therefore, the contribution of each forcing term to the wing response is determined as a function of γ_0 . To remain consistent with the small angle approximation, a range of yaw amplitude $0 \leq \gamma_0 \leq 0.3$ is considered. Equation 13 is solved independently for the Coriolis, Euler and centrifugal forces over the entire range of γ_0 assuming zero initial conditions (e.g. $q_k(0) = 0, \dot{q}_k(0) = 0$). The square sum of each partial response is divided by the square sum of the total response to determine the contribution of each forcing term as a percentage. The results of the simulation are shown in Figure 6.

At the operation point of γ_0 described in Table 5, the con-

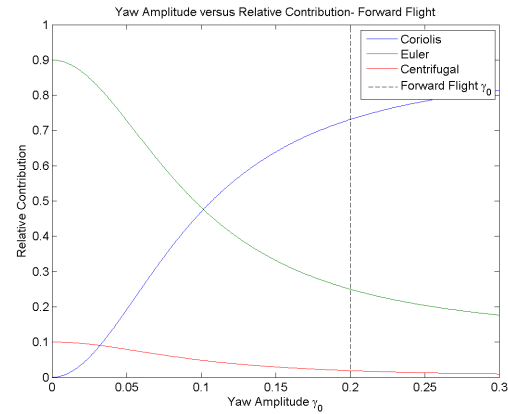


Figure 6: Relative Contribution of Forcing terms to Response as a function of γ_0

tribution of the Coriolis force to the total response is substantially larger than the contribution from the other two forces. In the forward flight regime, even large variations of γ_0 will cause only a minor change in the Coriolis contribution. Thus, an increase of γ_0 results in the amplification of the beat phenomenon. This study also shows the beat phenomenon can serve as an effective way to obtain $\dot{\gamma}$ from the wing response.

4.2 Sensitivity to Initial Conditions

The beat phenomenon results from an interaction between the forced vibration of the Coriolis force and the free vibration of the wing. From a purely mathematical standpoint, there exists a set of initial conditions that will cause the free vibration of the wing to disappear from the response. If there is no free vibration of the wing to interact with the forced vibration, the beat phenomenon will no longer be present.

Consequently, it is prudent to look at the Coriolis contribution as a function of non-zero initial conditions. To identify an allowable set of initial conditions, Equation 13 is solved assuming zero initial conditions. For the forward flight regime, the maximum and minimum values of $q(t)$ and $\dot{q}(t)$ constitute the upper and lower bounds of allowable initial conditions. Mathematically, this takes the form $q(t)_{min} \leq q(0) \leq q(t)_{max}$ and $\dot{q}(t)_{min} \leq \dot{q}(0) \leq \dot{q}(t)_{max}$. The Coriolis contribution is then determined by solving Equation 13 over the entire set of initial conditions using the time-dependent formulation. The results are shown in Figure 7.

Figure 7 shows the contribution of the Coriolis force is more than 60% for most initial conditions. Even for extreme initial conditions, the contribution of the Coriolis force still constitutes about 33% of the total contribution. Thus, some non-zero initial conditions may cause the beat phenomenon to become less distinct, but it will never entirely be removed from the total response.

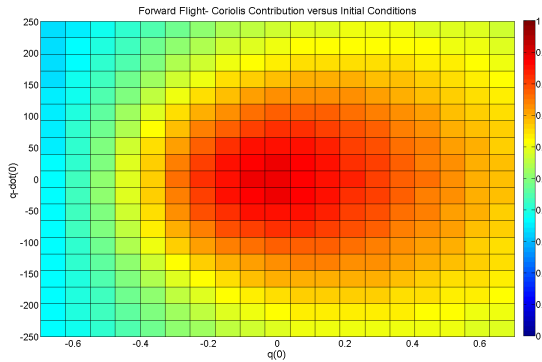


Figure 7: Relative Contribution of Coriolis Response to Various Initial Conditions

4.3 Damping

As the beat phenomenon relies partially on the free vibration of the wing, structural damping may have the effect of diminishing the beat over time. To determine the significance of these effects, structural damping of 1% ($\zeta = 0.01$) is introduced into Equation 13. The response subject to Coriolis forcing is then determined numerically for the first mode shape (Figure 8). As the free vibration of the wing dissipates as a result of damping, the beat phenomenon gradually becomes less pronounced. After several seconds, the Coriolis force causes only the forced vibration of the wing, which oscillates sinusoidally at $2\omega_d$.

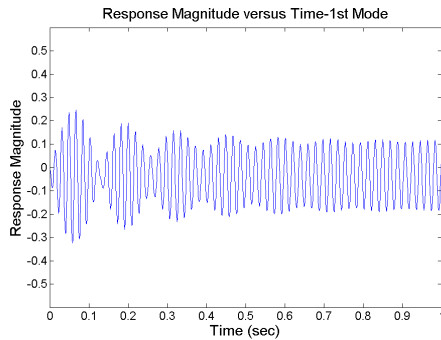


Figure 8: Damped generalized coordinate response to Coriolis forcing, First mode

For insects, there exist a number of ways in which the beat phenomenon may be preserved in the response in spite of structural damping. First, rotation amplitudes may be randomized slightly from their mean values to better represent flapping of a real insect. Subtle variations, particularly in the yaw amplitude, allow the beat phenomenon to be maintained. Second, the insect may perform an aerial maneuver, corresponding to both a considerable change in rotation amplitudes and an abrupt translation of the insect body. Both of these fac-

tors can cause free vibration to occur, thereby reestablishing the beat phenomenon. This suggests that it is desirable for an insect to move frequently to preserve neural feedback generated by deformation of the forewings.

4.4 Strains

Consider a point of interest on the wing $P(x, y)$. Let ϵ_x be the normal strain in the x-direction, ϵ_y be the normal strain in the y-direction, and τ_{xy} be the shear strain in the xy plane at $P(x, y)$. The strain components can be represented by

$$\epsilon_x(x, y, t) = \sum_{k=1}^{\infty} \epsilon_{x,k}(x, y) q_k(t) \quad (16)$$

$$\epsilon_y(x, y, t) = \sum_{k=1}^{\infty} \epsilon_{y,k}(x, y) q_k(t) \quad (17)$$

$$\tau_{xy}(x, y, t) = \sum_{k=1}^{\infty} \tau_{xy,k}(x, y) q_k(t) \quad (18)$$

where $\epsilon_{x,k}$ is the normal strain in the x-direction of the k^{th} mode, $\epsilon_{y,k}$ is the strain in the y-direction of the k^{th} mode, and $\tau_{xy,k}$ is the shear strain in the xy plane of the k^{th} mode. The first mode is a spanwise bending mode, resulting in significant strain at the center of the wing base. The second mode is a spanwise torsion mode, and results in large strains at the leading and trailing edge of the wing base.

A point $P(x, y)$ is selected near the base of the wing. The strain components are determined by finite element analysis for the first two mode shapes as shown in Table 6.

Strain component	First Mode ϕ_1	Second Mode ϕ_2
ϵ_x	256	-161
ϵ_y	-1577	96.7
τ_{xy}	691	1710

Table 6: Strain components normalized with respect to mass matrix for first two modes at $P(x, y)$

$q_1(t)$ and $q_2(t)$ are calculated by numerically solving Equation 13 assuming zero initial conditions and using the $\vec{a}_k$ and $\vec{b}_k$ vectors. The total time-dependent strain is determined using the strain values from Table 6 and applying Equations 16-18. The results are shown in Figure 9. The shear strain τ_{xy} has the largest magnitude, while the x-component of strain ϵ_x and the y-component of strain ϵ_y have comparatively negligible magnitudes. For all strain components, the beat phenomenon is observed.

5 CONCLUSION

A comprehensive dynamic model of an insect wing is developed treating the wing as a deformable body subject to three dimensional rotation about a fixed point on the insect's body. A coordinate frame that rotates with the finite pitch

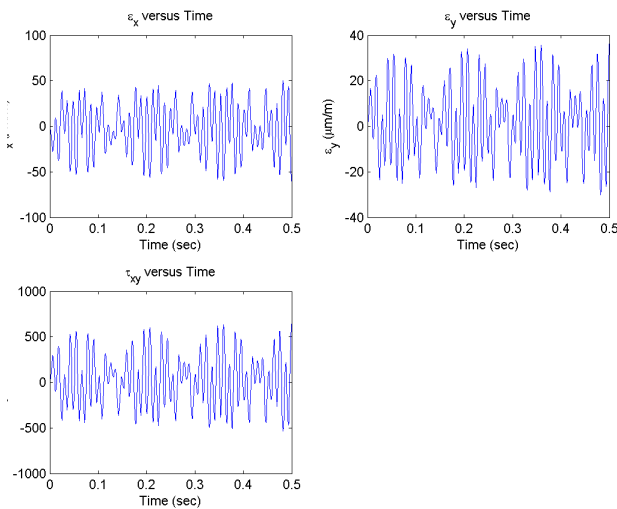


Figure 9: Time dependence of strain components at $P(x, y)$

and roll rotations is established. The yaw rotation is assumed small, and is allowed to rotate independently of the coordinate frame. The kinetic and potential energies are formulated, and the set of uncoupled differential equations describing the time response of each mode shape (determined by finite element analysis) is determined by Lagrange's equation. There are several advantages to such a formulation. First, the model is able to estimate the time-dependent strain components on the wing. Second, the model allows any complex geometry to be implemented readily. Third, the resulting model has reduced order—the order of the model is the number of mode shapes retained.

The model is then applied to the forewing of the *Manduca sexta*. Coriolis, Euler, and centrifugal forces are identified excitation terms responsible for the elastic deformation of the wing. The Coriolis force tends to dominate the first mode whereas the centrifugal force tends to dominate the second mode. Numerical integration indicates a beat phenomenon occurs in both the first and second modal responses. The sensitivity of the beat phenomenon in the fundamental mode is explored. The beat phenomenon is relatively insensitive to various yaw amplitudes and non-zero initial conditions, but decays gradually in the presence of damping. Resetting of initial conditions or slight variations of the yaw amplitude cause the beat phenomenon to persist even in the presence of damping.

The total time-dependent normal and shear strain components are calculated at a point of interest near the base of the wing. The shear strain component is determined to be significantly larger than the normal strain components. In all cases, the strain components exhibit a beat phenomenon. This result suggests the *Manduca sexta* may use strain frequency data associated with the beat phenomenon to help identify flight

status. In the context of MAVs, similar strain-based control systems may be developed to replace or supplement traditional accelerometer-based control systems. This can potentially increase MAV performance by reducing weight, size, and power consumption.

6 REFERENCES

- [1] *An Introduction to Fluid Dynamics*. Cambridge, 1967.
- [2] T.L. Daniel and S.A. Combes. Into thin air: contributions of aerodynamic and inertial-elastic forces to wing bending in the hawkmoth *manduca sexta*. *The Journal of Experimental Biology*, 2003.
- [3] Ron Galant, James B. Skeath, Steve Paddock, David L. Lewis, and Sean B. Carroll. Expression pattern of a butterfly *achaete-scute* homolog reveals the homology of butterfly wing scales and insect sensory bristles. *Current Biology*, 1998.
- [4] R.C. Michelson and S. Reece. Update on flapping wing micro air vehicle research. In *Proceedings of 13th Bristol International RPV Conference*.
- [5] Adam P. Tobias. *Experimental methods to characterize nonlinear vibration of flapping wing micro air vehicles*. PhD thesis, Air Force Institute of Technology, 2007.
- [6] R.J. Wood. Liftoff of a 60mg flapping-wing mav. In *Intelligent Robots and Systems*.
- [7] Bruno A. Roccia, Sergio Preidikman, Julio C. Massa, and Dean T. Mook. Development of a kinematical model to study the aerodynamics of flapping-wings. *International Journal of Micro Air Vehicles*, 2011.
- [8] Christopher T. Orlowski and Anouck R. Girard. Modeling and simulation of nonlinear dynamics of flapping wing micro air vehicles. *AIAA Journal*, 2011.
- [9] Satish K. Chimakurthi and Carlos E. S. Cesnik. Flapping-wing structural dynamics formulation based on a corotational shell finite element. *AIAA Journal*, 2011.
- [10] Alexander P. Willmott and Charles P. Ellington. The mechanics of flight in the hawkmoth *manduca sexta*. *The Journal of Experimental Biology*, 1997.
- [11] Aaron G. Norris, Anthony N. Palazotto, and Richard G. Cobb. Experimental structural dynamic characterization of the hawkmoth (*manduca sexta*) forewing. *International Journal of Micro Air Vehicles*, 2013.
- [12] Mark A. Frye. Encoding properties of the wing hinge stretch receptor in the hawkmoth *manduca sexta*. *The Journal of Experimental Biology*, 2001.

Design, Fabrication, and Performance Test of a Hovering-Based Flapping-Wing Micro Air Vehicle Capable of Sustained and Controlled Flight

Quoc-Viet Nguyen *, Woei Leong Chan, and Marco Debiasi
National University of Singapore

ABSTRACT

We present our insect-based hovering Flapping-Wing Micro Air Vehicle (FW-MAV) capable of sustained flight and control. The proposed design combines two fixed wings and two flapping wings to take advantage of the double clap-and-ling effects at the end of each half-stroke during one flapping cycle for high thrust production, and utilizes the fixed wings as the stabilizing surfaces and lift enhancement mechanism in forward flight. The FW-MAV has a wing span of 22 cm from wing tip-to-wing tip, weighs about 14.6 grams with onboard integration of radio control system including a radio receiver, an electronic speed control (ESC) for brushless motor, two servos for attitude flight controls of pitch and yaw, and a single cell lithium polymer (LiPo) battery (3.7 V). Moreover, passive wing rotation is utilized to simplify the design. Powered by the single cell LiPo battery, the FW-MAV can produce enough vertical thrust of 14.76 grams for lift-off at 10 Hz, and flap at maximum frequency of 12.4 Hz with average vertical thrust of about 24 grams. Experimental results show that the effect of wing clap significantly enhances thrust generation of about 44.82% when compared to the case without clap-and-fling effect, and the thrust-to-power ratio of the FW-MAV is about 3.93. Finally, free flight tests in terms of vertical take-off, hovering, and manual attitude control flight have been conducted to verify the flight performance of the insect-inspired FW-MAV. Onboard equipped with a 70 mAh LiPo battery, the FW-MAV can demonstrate a sustained and controlled flight of about three minutes.

1 INTRODUCTION

NOWADAYS flapping-wing micro air vehicles (FW-MAVs) become more and more attractive flying model to many research scientists around the world since they are well-known for effectively flight performance by employing many unsteady aerodynamic mechanisms in a low range of Reynolds number, which is usually less than 10,000 [1], and agile maneuverability compared to their counter parts of fixed wing and rotary wing for indoor operation. Based on natural inspiration of bird flight and insect flight, man-made FW-MAVs have been split into two main categories: bird-inspired FW-MAVs and insect-inspired FW-MAVs. They both fly by flapping wings, but fundamentally differ from each other in many ways [2] in terms of design of wing kinematics and control mechanisms. Most of bird-inspired

FW-MAVs known as ornithopter need an initial launching speed while flapping their wings to produce propulsion and lift to push them forward and to keep them aloft in air. Thus, they cannot take off vertically or hover. In contrast, one of the ultimate goals of insect-inspired FW-MAVs is hovering capability. For flight, an insect-inspired FW-MAV should flap its wings in a nearly horizontal plane [3] with a large flapping stroke angle and wing pitch to produce sufficient vertical thrust even without forward velocity, allowing them to fly at low forward speed or even hovering without stalling. The hover capability and potential high flight efficiency are the two key factors that FW-MAVs still draw researchers' attention. However, we might acquire an efficient FW-MAV only if we can properly mimic principles of nature's flyers such as birds and insects.

Recently, countless FW-MAVs have been developed and flown by both academia and industry, only a few FW-MAVs are able to demonstrate free hovering flight [4-12]. Perhaps it is due to the difficulty to cope with the inherent instability of the hovering-based flapping-wing platforms [13]. In 2009, the Nano Hummingbird [4] of AerovironmentTM (AV) became the only FW-MAV that met all of the Phase II technical milestone set out by DARPA [14]. Recently, TechJect [5] and Festo [6] have also announced their dragon-like four-winged FW-MAVs capable of hovering in 2012. Apart from the industries, the academia has also brought forth a few notable platforms [7-9] capable of hovering with onboard control system.

In this paper, we describe a hybrid insect-inspired design between flapping wing and fixed wing, and fabrication of a FW-MAV capable of vertical take-off and hovering. The FW-MAV shown in Figure 1 has a wing span of 22 cm from wing tip-to-wing tip, weighs less than 20 grams, and mimics some desirable features of insect flight such as large wing stroke angle, wing pitch angle, and clap-and-fling mechanism for high thrust production. Keeping in mind of natural inspiration of insect flight, we designed a gearbox mimicking an insect thorax by using four-bar linkages, in which clap-and-fling effects can be created at the end each half stroke: upstroke where the two flapping wings clap together at dorsal side (dorsal wing clap), and downstroke where the two flapping wings clap on the two fixed wings placed at the ventral side

*Email address: tslnqv@nus.edu.sg

(ventral wing clap) of the FW-MAV. As shown in Figure 1, four wings are arranged around the fuselage, but only two wings flap, the other two wings are fixed. Thus, the platform is a hybrid between flapping wing and fixed wing. The fixed wings play main roles of creating clap-and-fling effect at ventral side for vertical thrust enhancement, stability, and lift enhancement for forward flight mode. In addition, passive wing pitch is incorporated into the FW-MAV by means of an initial slack angle implemented on the wing, and cylindrical hinges placed at the wing root. For stability and control, we still use the bird-like tail and control surface. Attitude control flight of pitch and yaw are performed by control surfaces (elevator and rudder) actuated by two servos. Finally, force measurement and free flight test allow us to characterize the performance of the FW-MAV in terms of vertical thrust production, dynamic flight stability, and control effectiveness; the FW-MAV can vertically take off at flapping frequency of 10 Hz, and carry a maximum payload of 9 gram at maximum flapping frequency of 12.4 Hz. Hovering flight can be easily achieved by control surface (elevator).

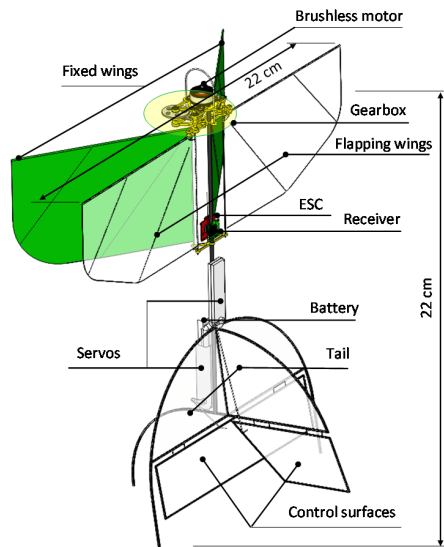


Figure 1: Three-dimensional SolidWorks model of the FW-MAV.

2 BIOMIMETIC DESIGN OF THE FW-MAV

With the inspirations from natural flyers, the design principle of the FW-MAV is to mimic some novel features of insect and hummingbird flights which can enable the FW-MAV to take off vertically and hover in air. Therefore, the flapping wings need to be configured in such a way that they provide enough vertical thrust for hovering flight.

2.1 Flapping mechanism

The foremost and most important thing we have to consider is the flapping mechanism used to drive the wings. For that purpose, we looked at an insect's wing joint system as shown in Figure 2, and mimicked it by using four-bar linkages (crank-rocker mechanism). This mechanism is

simple and has one degree of freedom; each bar connects to two other bars by cylindrical joints as shown in Figure 3.

In order to drive the four-bar mechanisms, we use a gear system in which input links/cranks are fixed to the reduction gears (Figure 3). Thus, the rotary motion of the motor is directly transformed into the rotary motion of the input links. Sequentially, the four-bar mechanisms transform the rotary motion of the input link into the reciprocating motion of the output links/rockers where the wings are attached. In this design, we choose a flapping stroke angle of 130° inspired from insect flight [1], and reduction gear ratio of 1 to 16 to utilize the maximal motor power at the half maximal speed. Figure 4 shows the position angle of the output link from the position analysis of the four-bar linkage under no load condition [15]. The mean flapping stroke angle is intentionally shifted to the dorsal side in order to create a clap-and-fling effect at the end of upstroke where the two flapping wings clap together. For accommodation for the flapping mechanism, we assembled a gearbox, Figure 5, in which we used 1 mm high gloss carbon fiber sheet to fabricate all the gearbox housing and links by mean of milling computer numerical control (CNC) machine (M300S CE, Woosung E&I Co. Ltd, Korea). The details of fabrication process can be found in the reference [9].

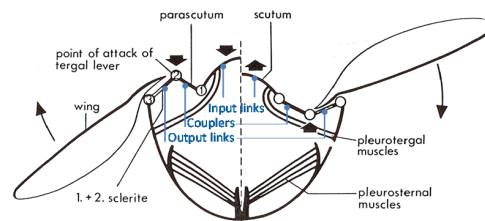


Figure 2: Wing joint system of an insect adapted from [16].

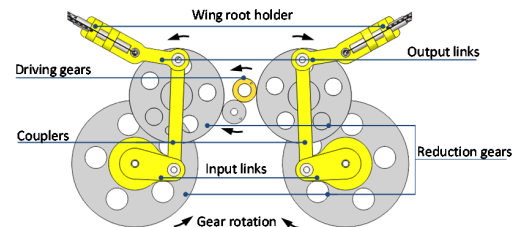


Figure 3: Gears and four-bar linkages mimicking insect's thorax.

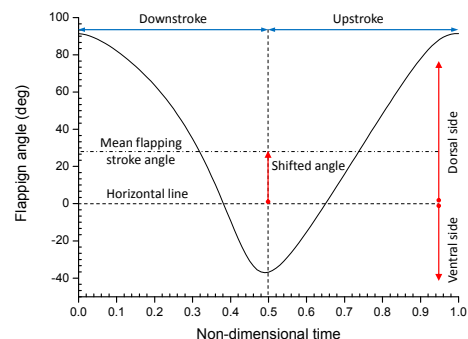


Figure 4: Position of flapping stroke angle of the output link.

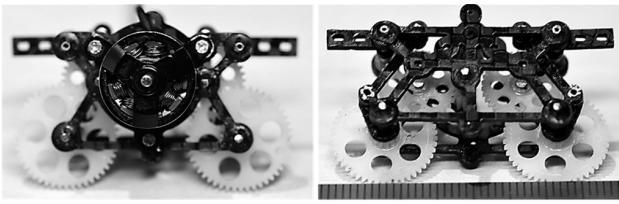


Figure 5: Full assembly of the gearbox (one brushless motor, two pinions, two 36/12 teeth spur gears, two 48/12 teeth spur gears, two four-bar mechanisms, and housing frames).

2.2 Wing and wing pitch mechanism

The flapping wings should be flexible to produce high thrust [17], but stiff enough to sustain the dynamic wing loading during flapping, and as light as possible to reduce the undesirable effect of wing inertia at high flapping frequency. For this purpose, we use lightweight but high strength materials: thin carbon rod (0.8 mm for leading edge and 0.3 mm for wing supporting veins) and thin Mylar film (15 μ m) for membrane. The supporting veins are intentionally not joined with wing leading edge (WLE) allowing the wing to easily deform for wing rotation about WLE while maintaining the wing shape for aerodynamic force generation during flapping motion. Figure 6 shows a fabricated wing with the area of 61 cm², and the weight of 0.7 gram. The implemented initial slack angle, θ , is for passive wing pitch/rotation.

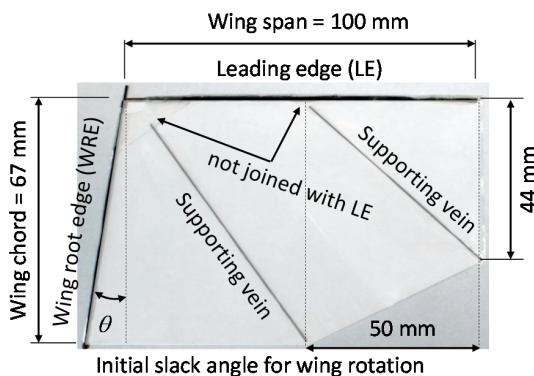


Figure 6: Wing made of carbon rods and Mylar film.

Insects fly by sweeping their wings at large flapping stroke angle and actively pronating and supinating their wings at the end of each half stroke. These active wing pronation and supination adjust the wings to a right angle of attack in order to capture vortices from the previous stroke for lift/thrust enhancement. Mimicking such an active wing pitch is a challenging part of design and requires a complex mechanism that may lead to weight penalty. Instead of that, a passive wing pitch mechanism induced by resultant aerodynamic forces and wing inertia is a simple and effective mechanism, and mostly used in current FW-MAVs. The feathering/rotational axis is located at the wing LE. When the wing flaps, resultant aerodynamic force on the wing makes it passively pitch about the feathering axis in clockwise (CW) or counter-clockwise (CCW) directions, as shown in Figure

7, resulting in vertical thrust production for both downstroke and upstroke. In order to prevent excessive wing pitch angle, we intentionally implement an initial slack angle, θ in Figure 6, on the wing to create an initial wing deformation when it is installed into the FW-MAV where the wing root edge (WRE) is perpendicularly aligned with the wing LE. Moreover, we utilize cylindrical hinges placed at the wing root, Figure 8, to allow free rotation of WLE and WRE for more effectively passive wing rotation. The maximum clock-wise (CW) and counter clock-wise (CCW) wing pitch angle, ψ in Figure 5, at wing tip is designed to be 30° for the downstroke and 30° for the upstroke, respectively, by implementing an initial wing slack angle of 5°; this angle was experimentally proved to be the best wing configuration for maximum lift production among various initial wing slack angles from 0° to 15°.

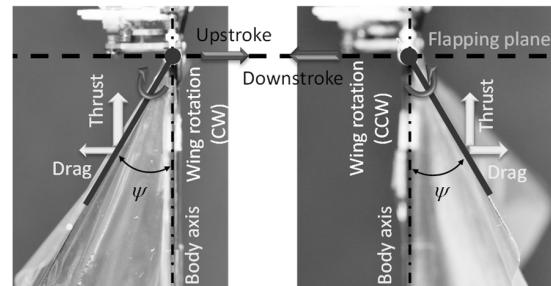


Figure 7: Passive wing pitch during downstroke and upstroke.

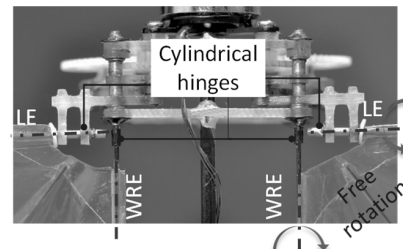


Figure 8: Cylindrical hinges at wing root for effectively wing rotation.

2.3 Wing clap-and-fling mechanism

Clap-and-fling mechanism proposed by Weis-Fogh [18], analyzed to confirm the existence and power of the concept by Lighthill [19], and quantitatively evaluated by Bennett [20] is one of the methods of insect flight along with the leading edge vortex method [21]. This mechanism allows the flapping wings to rapidly build and increase circulation due to absence of trailing edge vorticity [22], resulting in high lift/thrust production. Figure 9 describes the sequence of the clap-and-fling mechanism.

In this design, we mimic the insect wing clap-and-fling mechanism by designing a proper shape for the output links, where the flapping wings are attached, such that the wing claps can be created at the end of each half stroke: the end of upstroke where the two flapping wings clap together (dorsal wing clap), and the end of downstroke where the two flapping wings clap on the two fixed wings (ventral wing claps). Figure

10 shows the flapping angle and implementation of wing claps into the FW-MAV. The effect of the wing claps on vertical thrust production was experimentally investigated in the section 3. Figure 11 indicates the wing claps captured by a high speed camera.

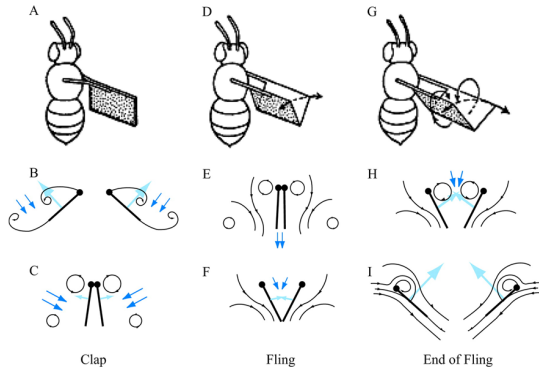


Figure 9: Clap-and-Fling mechanism proposed by Weis-Fogh and modern view adapted by SP. Sane [19].

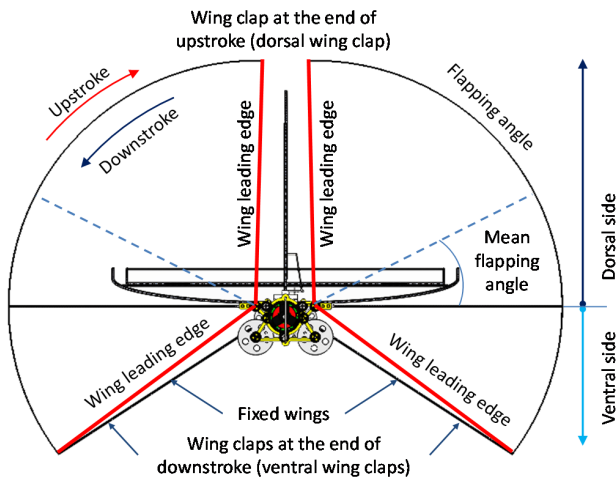


Figure 10: Top view: flapping stroke angle and wing claps at dorsal side and ventral side of the FW-MAV.

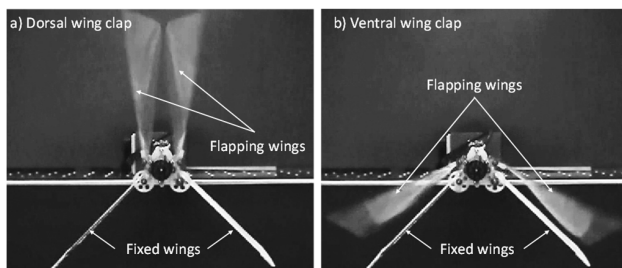


Figure 11: Wing claps at dorsal and ventral sides captured by a high speed camera.

2.4 Control mechanisms and stability

We use bird-like tail for stability and the conventional control surfaces as those in an airplane to simplify the design and perform attitude control of pitch and yaw for the FW-

MAV. Figure 12 shows the control mechanism integrated on our present FW-MAV; the two control surfaces which are elevator and rudder are tilted by two servos. The elevator can be tilted up and down to create a pitching moment to rotate the FW-MAV about the Y-axis in CCW or CW direction, respectively. Similarly, the rudder can be tilted left and right to create a yawing moment to rotate the FW-MAV about the Z-axis. The two degrees of control allow us to effectively control the FW-MAV. Hovering and forward flight are performed by the elevator, and turning flight is performed by the rudder.

For stability of the FW-MAV, we implement inherent stability by using stabilizing surfaces which consist of the two fixed wings and tail as shown in Figure 1. By placing a proper location of center of gravity (CG), the FW-MAV can maintain good dynamic flight stability during free flight test performed in the section 4. Comprehensive stability analysis is not considered in this work.

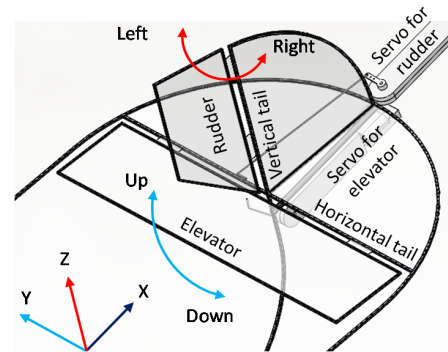


Figure 12: Tail and control mechanism.

2.5 Electronic system integration

The success of fabricating a controllable FW-MAV is somewhat determined by the availability of tiny size remote control components. Currently, the components used in the FW-MAV are all off-the-shelf. The receiver is the DelTang Rx31d compatible to Spektrum DSM2 protocol, weighs only 0.24 g, and it is among the smallest receiver for servo system available in the market. For motor, it is known that brushless motors provide higher efficiency at high speed operation compared to brushed motors. However, a brushless motor needs an electronic speed controller (ESC) to operate. One might consider that as an added weight, but we manage to find a 0.3 g ESC (HK030, HobbyKing.com) for the 3.1 g brushless motor (AP-03, kv = 7000, HobbyKing.com) which we are using in our FW-MAV. For control purpose, we use the 0.8 g Toki BioMetal servos (Smart Servo RC-1, positive PWM signal of $1.5\text{ms} \pm 0.5\text{ms}$ in every 20ms, www.toki.co.jp). Figure 13 and Figure 14 show the wiring diagram for the electronic system integration, and implementation of the electronic system on the fuselage, respectively. Figure 15 shows a complete assembly of the FW-MAV weighing 14.6 g whose detailed breakdown is shown in Figure 16 and parameters is listed in Table 1.

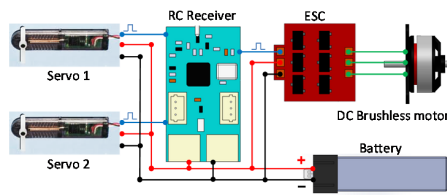


Figure 13: Wiring diagram for electronic components.

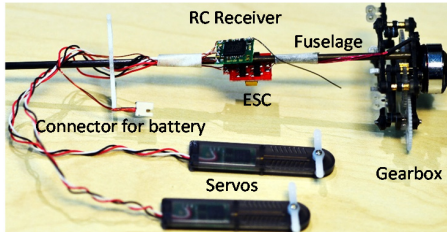


Figure 14: Electronic system integrated on the FW-MAV.

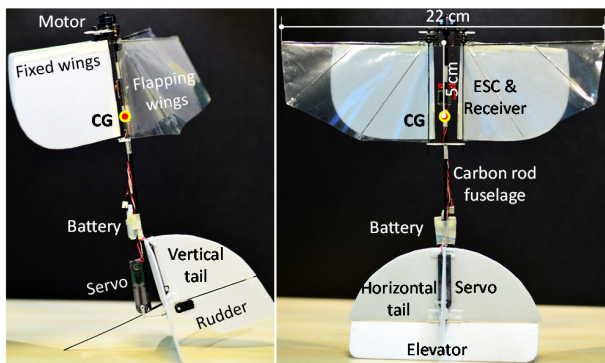


Figure 15: Full assembly of the FW-MAV.

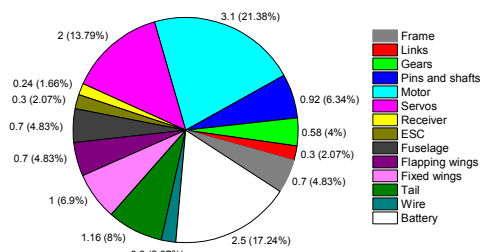


Figure 16: Weight breakdown of the FW-MAV in gram and percentage.

Parameters	Value
Mass	14.6 grams
Wing flapping frequency	10 Hz – 12.4 Hz
Take-off flapping frequency	10 Hz
Flapping stroke angle	125°
Wing length	10 cm
Wing span (wing tip-to-wing tip)	22 cm
Wing area (one wing)	61 cm ²
Wing loading	0.12 gram/cm ²
Wing aspect ratio	1.64
Mean chord length	6.1 cm
Mean wing tip velocity at 12.4 Hz	5.41 m/s
Reynolds number [1] at 12.4 Hz	10921

Table 1: Parameters of the FW-MAV.

3 FORCE PRODUCTION AND POWER CONSUMPTION

3.1 Experimental set-up

The force measurement was conducted to characterize the performance of the FW-MAV at various flapping frequency. Figure 17 shows the experimental set-up for force and power consumption measurement; the FW-MAV excluding tail and control mechanism was vertically mounted on a test jig consisting of a 3-axis force/torque loadcell (ATI Titanium Nano 17, ATI Industrial Automation), and a fixture. The loadcell can simultaneously sense six force components (3 forces and 3 moments), but only vertical force is considered in this test. An external power supply (Tenma 72-8350) was used in order to constantly maintain the applied voltage of 3.7 V to the motor. High voltages were refrained from applying because they are not compatible with the onboard electronic system. In order to reduce data contamination due to vibration of the FW-MAV itself and supporting structure, the sensing surface of the load cell was placed at the CG of the FW-MAV (Figure 15), and thus allowing us to obtain repeatable force data. To eliminate the undesired high peaks due to the vibration of the FW-MAV and structure vibration, we filtered the measured raw data by a low pass filter with a cut-off frequency that was 2.5 times greater than the flapping frequency. Thus, the filtered data included the effect of clap-and-fling and wing rotation frequency which happen two times in a flapping cycle. For each frequency, we carried out five tests for force measurement. Then, we took the cycle-average value of 20 cycles for each test and the statistical mean value of five tests to represent the average force generated by the FW-MAV.

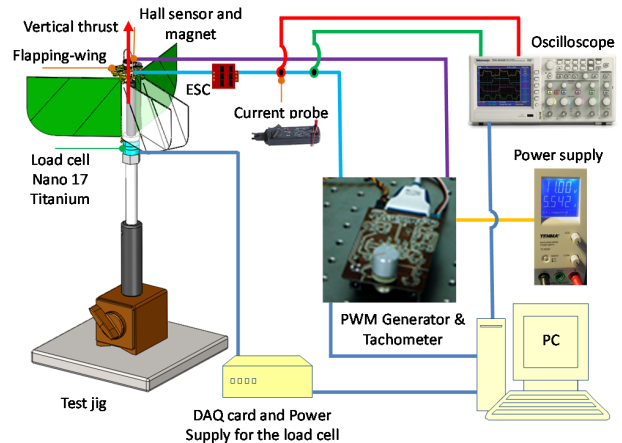


Figure 17: Experimental set-up for force and power consumption measurement.

To control flapping frequency of the FW-MAV, a Neodymium magnet is placed in a gear, where the input link is attached, and a Hall Effect switch is installed to pick up the magnetic signal from the magnet. Every time the magnet passes through the Hall Effect switch, an electrical pulse is generated and then transmitted to a tachometer circuit built in house, Figure 17. A micro-controller in the tachometer circuit

determines the flapping frequency and transmits the data to a computer via RS-232 port. The circuit also consists of a pulse width modulation (PWM) generator of which we can control the 1 ms to 2 ms pulse width by tuning the knob on the circuit, Figure 17. The PWM signal is transmitted to the electronic speed controller (ESC), and thus changes the motor speed, resulting in changing flapping frequency of the FW-MAV up to maximum value of 12.4 Hz.

Power consumption of the FW-MAV is calculated by voltage and current which were measured by an oscilloscope (TDS 2024B, Tektronix) and a current probe (CP-06 AC/DC, Kilter Electronic Institute) arranged as in Figure 17. The current probe sense the current flowing through a conductor and convert it to a voltage that can be viewed and measured on the oscilloscope. Channels 1 and 2 of the oscilloscope were used to read voltage applied to the motor, and voltage from the current probe, respectively. The voltage measured by the current probe was then converted into current by multiplying by the conversion factor indicated on the current probe.

3.2 Vertical thrust and effect of wing clap-and-fling

Force measurement was conducted for several flapping frequencies from 9 Hz to 12 Hz in which the FW-MAV produces enough thrust for take-off and flight. To investigate how the wing clap-and-fling affects vertical thrust generation, we conducted force measurement for three different flapping conditions: single wing only in which only one wing was installed into the flapping testing system, dorsal wing clap-and-fling in which two flapping wings clap together at the dorsal side, and dorsal & ventral wing clap-and-fling in which the two flapping wings clap together at the dorsal side (dorsal wing clap-and-fling) and clap on the two fixed wings at the ventral side (ventral wing clap-and-fling), respectively. The contributions of the wing clap-and-fling effects were examined by comparing the mean vertical thrusts produced by the FW-MAV for the three cases. Comprehensive investigation of this effect is out of this work scope, we leave it for future work.

The results shown in Figure 18 indicate the nearly linear relationship between flapping frequency and thrust in which the mean thrusts and standard deviations were statistically calculated from five measurements at each flapping frequency (9 Hz, 10 Hz, 11 Hz, and 12 Hz). Theoretically, the thrust is proportional to velocity or frequency square. Therefore, the nearly linear relationship obtained from the experiment can be explained by the induced velocity which reduces effective angle of attack; the induced velocity increases as the flapping frequency increases, resulting in reducing effective angle of attack. Thus, lift on the flapping wing is reduced.

At flapping frequency of 10 Hz the FW-MAV produces 14.76 grams thrust, which is slightly higher than its own weight of 14.6 grams, and starts to take off. At flapping frequencies of 11 Hz and 12 Hz, the FW-MAV can carry payloads of 3.3 grams and 7.2 grams, respectively, thus allowing us to implement a small camera onboard for the next version of the FW-MAV.

Wing clap-and-fling effects at dorsal and ventral sides were examined by comparing the two cases of dorsal wing clap-and-fling and single wing x 2 (which is calculated by multiplying the single wing case by 2), and the two cases of dorsal & ventral wing clap-and-fling and dorsal wing clap-and-fling, respectively. As shown in Figure 18, the wing clap-and-fling effect significantly increases the average vertical thrust. When compared to the case of single wing x 2, the dorsal wing clap-and-fling and dorsal & ventral wing clap-and-fling contribute to a thrust increase of 30.22% and 44.82%, respectively, at flapping frequency of 12 Hz. When compared to the case of dorsal wing clap-and-fling, the dorsal & ventral wing clap-and-fling produces a thrust increase of 1.73 grams (13.26% increase), 2.02 grams (12.68% increase), and 3.14 grams (16.83% increase) at frequencies of 10 Hz, 11 Hz, and 12 Hz, respectively. Therefore, it is worth to install the two fixed wings with weight of 1 gram to have higher thrust production.

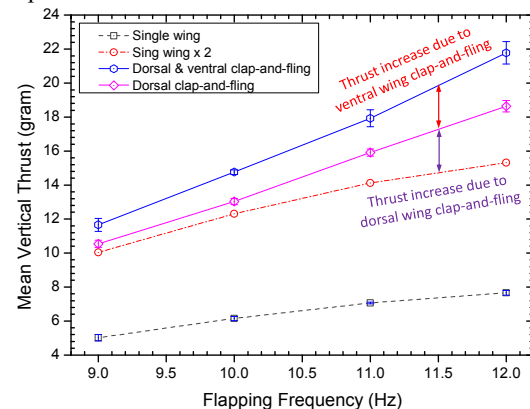


Figure 18: Mean vertical thrust and standard deviation for five measurements at each flapping frequency.

The measured thrusts indicate that the positive value dominates the thrust profile in both downstroke and upstroke. That means the passive wing rotation mechanism is properly implemented into the FW-MAV. The two peaks in the thrust profile/time-dependent thrust are due to the wing clap-and-fling effects at the end of each half flapping stroke: dorsal wing clap-and-fling and ventral wing clap-and-fling. The gaps between the time-dependent thrusts for the three cases of flapping condition at these two peaks indicate the wing clap-and-fling effects. As shown in Figures 19, 20, and 21 for three cases of flapping frequency (10 Hz, 11 Hz, and 12 Hz, respectively), the dorsal wing clap-and-fling creates a thrust increase at the higher peak (1st peak) when comparing the case of single wing x 2 to the cases of dorsal wing clap-and-fling and dorsal & ventral wing clap-and-fling, and the ventral wing clap-and-fling creates another thrust increase at the lower peak (2nd peak) when comparing the case of single wing x 2 and dorsal wing clap-and-fling to the case of dorsal & ventral wing clap-and-fling. We will further investigate these effects by the means of high speed cameras and smoke/He-bubble/PIV visualization in future to explain the high peak force in the time-dependent force profile. From the measured

results, we obviously expected that the FW-MAV can demonstrate vertical take-off and free hovering flight.

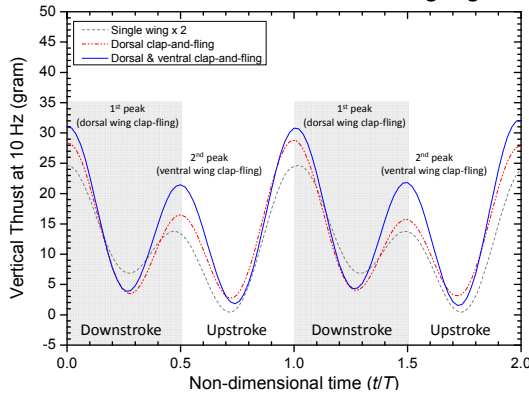


Figure 19: Time-dependent vertical thrust at 10 Hz.

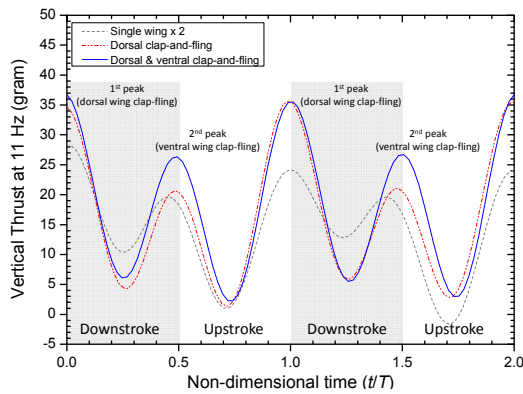


Figure 20: Time-dependent thrust at 11 Hz.

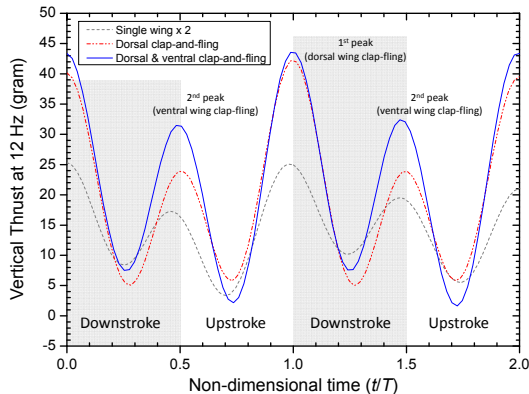


Figure 21: Time-dependent thrust at 12 Hz.

3.3 Power consumption

Power consumption was measured for two cases: gearbox only in which there was no wing installed into the FW-MAV, and full system (total power consumption) in which all wings were installed into the FW-MAV. Figure 22 shows the power consumption for the two cases from 9 Hz to 12 Hz. The power consumption due to the two flapping wings and wing inertia during flapping process was calculated by subtracting the power consumed by the gearbox only from the total power

consumption. At flapping frequency of 9 Hz, 10 Hz, 11 Hz, and 12 Hz, the power consumed by the gearbox only or power loss due to the gearbox is 18.75%, 17.41%, 17.04%, and 16.67% of total power consumption, respectively. However, in this test the power consumed by the wing inertia is still not isolated; power measurement in vacuum condition will have to be conducted in future. From the measurement of power consumption and thrust, thrust-to-power ratio turns out to be 3.32, 3.46, 3.68, 3.93, respectively.

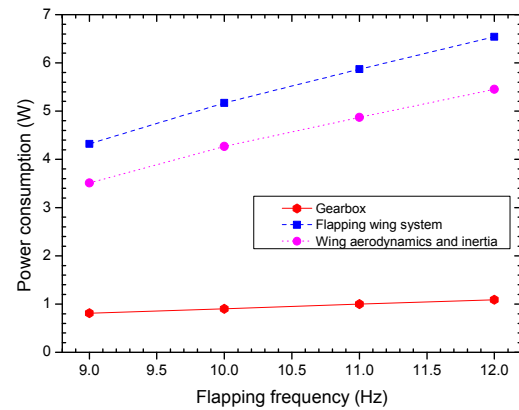


Figure 22: Power consumption of the FW-MAV at various flapping frequencies.

4 FREE FLIGHT TEST

Vertical take-off, sustained and controlled flight were conducted indoor with a radio transmitter (Spektrum DX7, DSM2) to demonstrate the flight abilities of the FW-MAV. The comprehensive study of stability and flight dynamics of the FW-MAV are not investigated in this work, we leave it to future work. The force measurement has proven that the FW-MAV produces sufficient thrust to lift off its own weight for flight. However, a visually real flight is much obviously to judge the force production of the FW-MAV. Thus, the take-off test was not only to verify the measured force but also to demonstrate the desired features of an insect-inspired FW-MAV: vertical take-off, and hovering. Moreover, the test was also to verify the inherent stability of the FW-MAV. Several take-off tests were conducted and captured by an ordinary camera, the captured were then extracted to build a composite image of take-off process. Figure 23 indicates the FW-MAV can demonstrate vertical take-off and then hovering.

In addition to the take-off test, free indoor flight test was also conducted to demonstrate and verify the sustained flight and control effectiveness, respectively. A simple mission was carried out as shown in Figure 24, which illustrates the flying sequence of the mission: take-off → hovering → attitude controlled flight → hovering → landing. Firstly, the FW-MAV vertically took off, and was tuned in hovering flight mode by using elevator. Transition flight from hover to forward flight/turning flight and vice versa could be achieved by both elevator and rudder. The mission lasting about one minute was successful accomplished. However, vertical

landing is still not achieved, the FW-MAV tends to be fallen down when it touches the ground. Thus, we will design a proper landing gear for the FW-MAV. Overall, our FW-MAV shows a good apparent performance and demonstrates a desirable feature of insect flight: hovering capability.

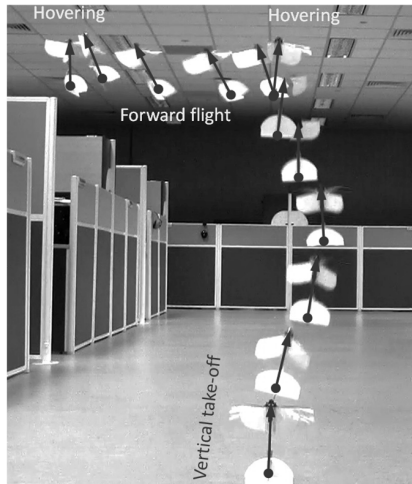


Figure 23: Vertical take-off and hovering, solid arrow denotes the body axis of the FW-MAV.

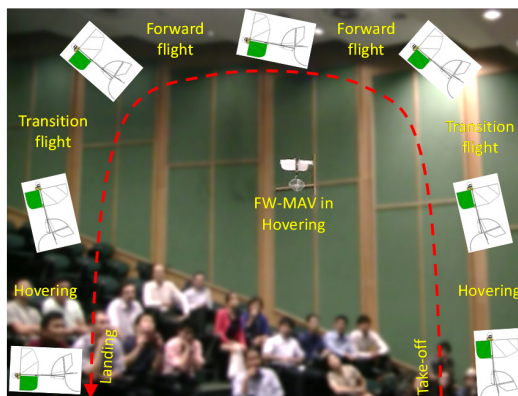


Figure 24: Flight demonstration.

5 CONCLUDING REMARKS

In summary, we presented a road map to achieve a hovering flapping-wing micro air vehicle mimicking principle of insect flight that involves translating a design concept into an actual flying flapping-wing prototype and then achieving the successful free flight with control authority. Our proposed hybrid design utilizes the clap-and-fling effects of insect flight for high thrust production that enable our 22 cm wing span hybrid insect-inspired FW-MAV to vertically take off and hover in air. Moreover, the measured vertical thrust of about 22 grams at applied voltage of 3.7 V and flapping frequency of 12 Hz indicates that the FW-MAV produces sufficient lift for hovering flight and payload carrying such as a system integration of a small camera for vision and a sensor for automatic control on the future versions of the FW-MAV. Future work will focus on further onboard system integration

for better control abilities, improving the FW-MAV such as implementation of compliant mechanism for energetic cost reduction, optimum wing design for high lift production, and unsteady flow visualization such as leading vortex mechanism, vortex shedding, clap-and-fling effect, and the generation of a downward flow for further performance improvement.

ACKNOWLEDGMENT

Authors appreciate the financial support from Defence Research & Technology Office of Singapore (Programme No. PA 9011102539).

REFERENCES

- [1] C. P. Ellington. The novel aerodynamics of insect flight: Applications to microair vehicles. *J. Exp. Biol.*, 202: 3439–3448, 1999.
- [2] W. Shyy, H. Aono, C.K. Kang, and H. Liu. *An Introduction to Flapping Wing Aerodynamics*. Cambridge Aerospace Series, Cambridge University Press, 2013.
- [3] F.O. Lehmann and S. Pick. The aerodynamic benefit of wing–wing interaction depends on stroke trajectory in flapping insect wings. *J. Exp. Biol.*, 210: 1362–1377, 2007.
- [4] Nano Hummingbird, available at <http://www.avinc.com/nano>
- [5] The Robot Dragonfly, available at <http://www.techject.com/>
- [6] BionicOpter–Inspired by dragonfly flight, available at http://www.festo.com/cms/en_corp/13165.htm
- [7] 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. *J. Aircraft*, 44 (5): 1701–1711, 2007.
- [8] <http://www.delfly.nl/home.html>
- [9] Q.V. Nguyen, W.L. Chan, and M. Debiase. Development of an insect-inspired flapping-wing micro air vehicle capable of vertical take-off and hovering. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Chicago, Illinois, Sept. 14–18, 2014. submitted for publication
- [10] C. Richter, H. Lipson. Untethered hovering flapping flight of a 3D-printed mechanical insect. *Artificial Life*, 17: 73–86, 2011.
- [11] F. van Breugel, W. Regan, and H. Lipson. From insects to machines: A passively stable, untethered flapping-hovering micro-air vehicle. *IEEE Robot. Autom. Mag.*, 15 (4): 68–74, 2008.
- [12] L. Ristroph, S. Childress. Stable hovering of a jellyfish-like flying machine. *J. R. Soc. Interface*, 11, 2014.
- [13] Y.L. Zhang and M. Sun. Control for small-speed lateral flight in a model insect. *Bioinsp. Biomim.*, 6, 2011.
- [14] “AV Achieves Technical Milestone: controlled hovering flight of NAV with two flapping wings,” AeroVironment Press Release, June 30, 2009.
- [15] D. B. Doman, C. P. Tang, and S. Regisford. Modeling interactions between flexible flapping wing spars, mechanisms, and drive motors. *J. Guidance, Control, and Dynamics*, 34(5): 1457–1473, 2011.
- [16] G.J. Goldsworthy, and C.H. Wheeler. *Insect Flight*. Boca Raton, Florida: CRC Press, Inc., 1989.
- [17] M. Vanella, T. Fitzgerald, S. Preidikman, E. Balaras, and B. Balachandran. Influence of flexibility on the aerodynamic performance of a hovering wing. *J. Exp. Biol.*, 212: 95–105, 2009.
- [18] T. Weis-Fogh. Unusual mechanisms for the generation of lift in flying animals. *Scient. Am.*, 233: 80–87, 1975.
- [19] M. Lighthill. On Weis-Fogh mechanism of lift generation. *J. Fluid Mechanics*, 60: 1–17, 1973.
- [20] L. Bennett. Clap and fling aerodynamics - An experimental evaluation. *J. Exp. Biol.*, 69: 261–273, 1977.
- [21] C.P. Ellington, C. van den Berg, A.P. Willmott, and A.L.R. Thomas. Leadingedge vortices in insect flight. *Nature*, 384 (6610): 626–630, 1996.
- [22] S.P. Sane. Steady or unsteady? Uncovering the aerodynamic mechanisms of insect flight. *J. Exp. Biol.*, 214 (3): 349–351, 2011.

Twirre: Architecture for autonomous mini-UAVs using interchangeable commodity components

J. van de Loosdrecht,* K. Dijkstra, J.H. Postma, W. Keuning and D. Bruin
NHL University of Applied Sciences, Center of Expertise Computer Vision,
P.O. Box 1080, 8900 CB, Leeuwarden, The Netherlands

ABSTRACT

Twirre is a new architecture for mini-UAV platforms designed for autonomous flight in both GPS-enabled and GPS-deprived applications. The architecture consists of low-cost hardware and software components. High-level control software enables autonomous operation. Exchanging or upgrading hardware components is straightforward and the architecture is an excellent starting point for building low-cost autonomous mini-UAVs for a variety of applications. Experiments with an implementation of the architecture are in development, and preliminary results demonstrate accurate indoor navigation.

1 INTRODUCTION

Nowadays, there is increasing interest in using mini unmanned aerial vehicles (UAVs) in a multitude of applications. For example, in the Netherlands, UAVs can be used for the inspection of agricultural lands, reviewing annual ditch cleanings, or the inspection of wind turbine blades. The market of UAVs is fast-growing and new hardware, software and applications quickly emerge. UAVs with cameras mounted on them can already be purchased as commodities.

Currently, however, in most applications the UAVs are controlled manually and the images are inspected by humans. Performing these tasks autonomously would make them less labor-intensive and thus more cost-effective. This involves autonomous flying of UAVs and automated (real-time) processing of recorded images to generate process or business support information.

This article focuses on such autonomous flight. Autopilots using GPS navigation are already a commodity [1, 2, 3], though autonomous systems for GPS-deprived environments are not. This is because complex sensor fusion and image analysis are needed to be able to cancel all dynamic effects during flight and to estimate the UAV's position in 3D. There are many UAV platforms available, often expensive and based on dedicated custom-built hardware and software.

In this paper the Twirre architecture will be introduced, which is a low-cost UAV platform designed for autonomous

flight. The platform should be able to perform missions autonomously from takeoff to landing, without assistance of manual control. The intention is to submit the system software to the public domain. To make the platform attractive for both experimental and commercial use, the following four design requirements have been kept in mind.

1. The platform consists of low-cost components. Many applications require costs to be minimized. A UAV will likely have a broader application range if the cost of the components are low. Furthermore, using autonomous UAVs always comes with a risk of material damage. For these reasons the UAV platform is composed of low-cost commodity mass-produced components, which can easily be exchanged, repaired or upgraded. Components like flight controllers and GPS modules can be purchased off-the-shelf.

2. The platform is upgradable and extendable. New versions of radio controlled (RC) hardware components and standard processor boards quickly become available. To allow for easy incorporation of new technology in the UAV, components should be upgraded or extended with ease. All software is developed in C(++) and therefore easily portable.

3. The platform is useful in multiple applications. For a UAV platform to be useful in multiple applications it should be flexible. This means that it should be possible to perform both GPS-enabled and GPS-deprived applications. For example, automated imaging of agricultural fields can be done with GPS-enabled flight. However, for inspecting wind turbine blades GPS is not accurate enough and information from additional sensors should be used to estimate the UAV's position. In order to process and interpret this information (including images) in real-time, without the aid of external resources, processing power is required on-board the UAV.

4. The platform should be able to switch instantly and reliably between manual and autonomous control. Regardless the application, safety is of great importance for autonomous flight. Therefore, in autonomous flight, a reliable method for reinstating manual control must be devised.

This paper starts by presenting a short overview of related work in Section 2, after which the Twirre architecture is introduced in Section 3. While the architecture can potentially

*Email address: j.van.de.loosdrecht@nhl.nl

be used for any type of UAV, Section 4 focuses on an example implementation on a hexacopter which can be used in GPS-deprived environments. Section 5 then shows the layout of several experiments that have been performed with this implementation. They are used to test the architecture. The results and their interpretation are given in Section 6. Section 7 contains a list of conclusions based on the implementation and the experimental results. The paper is wrapped up with a short elaboration on work still to be done in Section 8.

2 RELATED WORK

Shen et al. [4] describe an architecture similar to Twirre. However, Twirre focuses on using low-cost interchangeable commodity components, which is not a design requirement in their paper.

Most other research is focused on small aspects of autonomous flight, like landing [5], planning [6], or integration with specific hardware [7]. More complex tasks are often demonstrated in simulated environments only [8].

The Twirre architecture connects available hardware and software in a modular design, thereby filling the gap between technology and real-world autonomous applications. Research has been performed on popular components which can serve as an implementation for low-level and high-level control of the Twirre architecture. Low-level control encompasses basic flight control, whereas high-level control consists of intelligent navigation.

Low-level control (leveling, maintaining altitude) For low-level control a flight controller is used. Such hardware and software components can be purchased off-the-shelf. Currently, for multicopters, the DJI Naza¹ is a popular choice among RC model pilots. The Paparazzi autopilot is an open source alternative which is mainly used in academic research [9].

The Paparazzi platform contains a safety pilot feature built into the software of the autonomous controller which can be used to switch to manual flight. However, the autonomous controller cannot be bypassed and thus there is no ‘galvanic’ (in hardware) separation between the RC control and the actuators. Although thoroughly tested, this architecture could potentially lead to unsafe situations.

High-level control (POI interaction) High-level control is used to locate and interact with a *point of interest* (POI, sometimes referred to as *landmark*). Using GPS is one solution for this type of control. A possible POI could then be a GPS location. However, low-cost GPS in itself is only accurate up to a few meters which limits the possible applications. The proposed solution for augmenting position information uses differential GPS [10] or the placement of artificial landmarks in the surroundings [11, 12]. Even though for research purposes

artificial landmarks can be useful, most actual applications revolve around markerless navigation.

When relying solely on image analysis in GPS-deprived applications, several control solutions are available. One of them is to use feature detection [4] or the scale invariant feature transform (SIFT), possibly combined with an omnidirectional camera [13], as used with unmanned ground vehicles (UGVs). While UGVs are stable and usually only navigate on a planar surface, UAVs are unstable and navigate in a 3D environment. For a UAV to keep a constant position, a POI on the ground could be used. The drift can then be calculated by analyzing two consecutive frames by an optical flow algorithm [14], which uses a pixel descriptor to detect where points have drifted to. Other solutions are to use laser odometry [4] or simultaneous localization and mapping (SLAM) [15, 16]. However, high-level controllers are generally computationally expensive, which is why UAVs often rely on wireless connections to an external computing station [17]. Since a platform which is independent of external resources is more flexible, the Twirre architecture uses on-board processing.

3 MATERIALS

In this section the components comprising the Twirre architecture are described. The first subsection introduces the general components in an abstract manner, after which the components making up an actual implementation of the platform based on a hexacopter are introduced.

3.1 The Twirre architecture

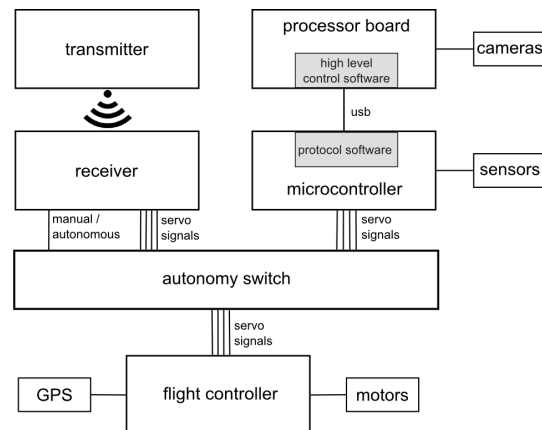


Figure 1: Twirre architecture.

Figure 1 shows the Twirre architecture consisting of six main components and several peripherals. There are two control pipelines. The left hand side shows manual control, while on the right hand side autonomous control is shown. The autonomy switch can be used to instantly switch from the autonomous pipeline to the manual pipeline. At the bottom of Figure 1 a stock flight controller is utilized for low-level con-

¹<http://www.dji.com>

trol in both manual and autonomous control, to provide the convenient flight control typically used by hobby RC UAVs. The flight controller translates stick inputs from the human pilot to motor inputs and can also be used for off-the-shelf GPS flight.

Manual pipeline The manual pipeline is almost identical to the ones used in manually controlled UAVs, with the only difference being the autonomy switch. The pipeline consists of a standard wirelessly connected transmitter/receiver pair. The servo signals are passed from the receiver through the autonomy switch to the flight controller. The manual to autonomous switching signal is controlled by a switch on the transmitter.

There is minimal software interference, and the software that is used in this process is the same software used in manual flight of a regular RC controlled UAV.

Autonomous pipeline The autonomous pipeline is dedicated to autonomous flight, based on on-board real-time processing of sensor data. Sensor fusion is used for interpreting surroundings and controlling the UAV. All sensor information is processed on the on-board processor board. The board can have several cameras connected to it, each possibly pointing in a different direction. The sensors connected to the microcontroller can provide additional information, which is sent over the USB bus to the processor board.

The software on the processor board processes the sensor information and translates it to commands which are sent over the USB bus to the microcontroller. The microcontroller translates these commands into standard servo pulse-width modulated (PWM) signals. These servo signals are sent to the flight controller through the autonomy switch.

Autonomy switch The autonomy switch provides a ‘galvanic’ separation between autonomous and manual flight. It gives a safe and instant method of taking control and is not controlled by the autonomous process. This gives the Twirre architecture a significant safety advantage compared to other architectures [9]. An additional virtue of this interface is that the UAV can be *partially* autonomous. For example, only the pitch of the UAV could run on the autonomous pipeline, while the remaining axes are manually controlled.

Cascade control system As follows from the distinction between low- and high-level control, Twirre utilizes a cascade control system. As mentioned before, for the low level a commodity flight controller is used. At the high level a control system is implemented in the software running on the processor board, which simulates the stick input from the human pilot. In autonomous mode, the mission specific software will determine the set-points for the high-level controller. By



Figure 2: The hexacopter. The diameter of the frame is 62 cm, or 85 cm including the protection ring.

splitting up the control system in two layers it is convenient to change or update to a new or better flight controller.

3.2 Components

The specific implementation of the Twirre architecture described in this paper is built around a standard DJI Flame Wheel 550 (F550) hexacopter which consists of a frame and six motors, electronic speed controllers (ESC) and propellers. For the flight controller a NAZA-M V2 flight controller is used. This platform is chosen because of its modular design and the ability to use GPS-enabled flights. The hexacopter is shown in Figure 2.

The processor board consists of a Commell LS-37B main-board with an Intel Core i7 3820QM processor with a clock speed of 2.7 GHz and a maximum turbo frequency of 3.8 GHz and 4 GB of DDR3 RAM. The processor board is running a standard Ubuntu 13.10 distribution. This high-end processing board makes it possible to perform on-board image analysis for autonomous flight.

Two industrial machine vision cameras are connected to the processor board. For the front camera an IDS UI-1241LE-C-HQ is used in combination with a 12 mm lens, whereas for the bottom camera an IDS UI-1221LE-M-GL is used in combination with an 8 mm lens. Industrial cameras are used because they have a global shutter. In a survey only consumer webcams using a rolling shutter were found, which are not suitable for image analysis applications with motion. To further stabilize the images the front camera is mounted on a servo controlled gimbal. This is shown in Figure 3.

The microcontroller board used is a MicroWii Flight Controller based on an ATmega32U4 microcontroller. This board is chosen mainly because it has all the necessary digital and analog in- and outputs. In this specific implementation it is not used as a flight controller, but it has a gyro / accelerometer (MPU-6050), a magnetometer (HMC-5883L) and a barometer (MS5611-01BA03) which can be useful during autonomous flight. An ultrasonic sensor (SRF08) is connected to the microcontroller for accurate altitude measure-

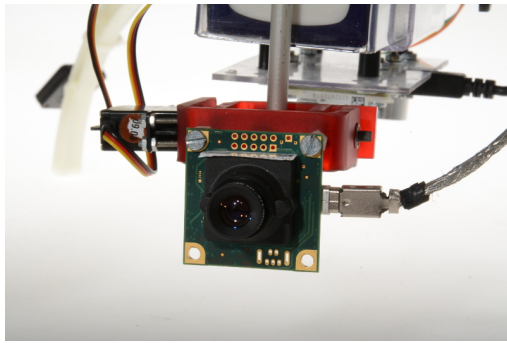


Figure 3: The IDS camera mounted on a servo controlled gimbal.

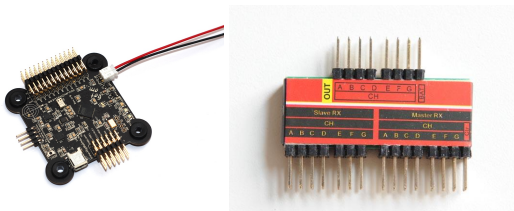


Figure 4: The MicroWii and the dual receiver controller.

ments.

The autonomy switch is implemented with a commodity \$15 RC dual receiver controller that is designed for training RC pilots. In its intended use, two receivers are connected to the input side, and the outputs are connected to the servos or flight controller. Both the student-pilot and the instructor have their own transmitter and receiver. The instructor can, with the use of a switch on his transmitter, switch between the receiver that is active. The used MicroWii and the dual receiver controller are shown in Figure 4.

For the transmitter and receiver a Graupner MX-12 and a Graupner GR-12, both operating in the 2.4 GHz band, are used. The dual receiver controller is manufactured by Assan.

VisionLab² is used as image processing library.

4 METHODS

Now that the components of the Twirre architecture have been introduced, it is time to look at the system software parts. The first subsection describes the protocol software on the microcontroller. After that, the second subsection introduces a test application that runs on the processor board, followed by subsections describing the test application in more detail.

4.1 Protocol Software

The protocol software has two functions:

1. Interpret commands received from the USB bus, convert these commands to standard servo PWM signals

²<http://www.vdlmv.nl>

and send them through the autonomy switch to the flight controller.

2. Read auxiliary sensor information, like gyro, barometer, etc. and send it over the USB to the processor board.

5

Figure 5: The POI being used in the test application.

4.2 Test application

In order to test the Twirre hardware architecture and the protocol software a basic UAV application is defined. In this application, the mission of the UAV is to search for an artificial POI, fly towards it and to face it. The POI being used is shown in Figure 5. It consists of a piece of A3 paper with a printed image of a solid black rectangular target with a white number inside.

There are many similarities between an autonomous UAV and UGV (Unmanned Ground Vehicle) performing this task. The main difference is when an UGV stands still, it actually stands still, whereas an UAV has the tendency to drift. The autonomous UAV flight discussed in this paper behaves like an UGV when searching for a POI. The test application software is a combination of the mission logic, input sensor data processing and the high-level control system.

4.3 Cascade PID controllers

The Twirre architecture uses cascade PID controllers to control the UAV. The low control level is embedded in the flight controller. The accelerometer, gyro and pressure sensor are used to keep the UAV leveled and at a constant height. When GPS can be used, the flight controller also controls the position of the UAV. These low-level controllers are commodity products. The high-level controllers are responsible for controlling the intelligent decision making of the UAV. In manually controlled UAVs high-level control is done by a human pilot.

The set-points for the high-level controller are determined by analyzing the surroundings and selecting a POI using the front camera and the processor board, and fly towards it. If there is no POI seen with the front camera, the bottom camera is used to keep a fixed position by estimating the drift using image analysis. In this case the yaw is used to rotate the UAV and scan the environment for new points of interest. The choice of bottom POI is made autonomously. The key is to always have a point of interest, whether seen from the front camera or from the bottom camera.

The Twirre architecture allows the UAV to be tested by mapping the pitch, yaw, roll and throttle to any combination of autonomous and manual control. This way individual, high-level controllers can be tested or several controllers can be tested simultaneously. This allows for detailed analysis of the autonomous behavior of the UAV. The autonomy switch

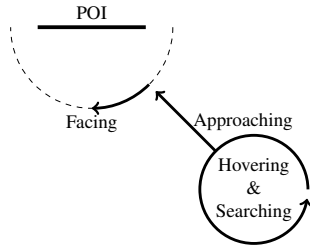


Figure 6: Top view of state transitions.

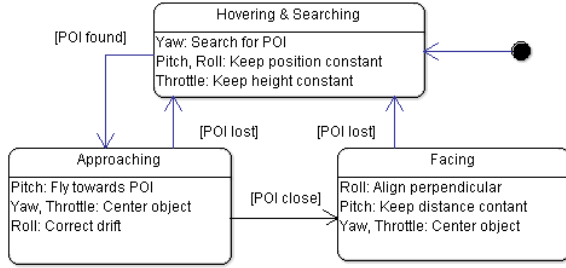


Figure 7: States of the high-level controller.

of the Twirre architecture can always be used to allow instant mapping to full manual control.

4.4 Controlling real-world parameters

Instead of focusing solely on image data, the Twirre architecture controls camera-independent, real-world parameters. So instead of using set-point deviations in pixels only, the platform relates these to real-world distances and angles, in millimeters and radians respectively. The advantage of this approach is that it allows for easy replacement and upgrading of cameras and lenses. Controllers are independent of the cameras and lenses being used. The PID controllers do not need to be re-tuned when a different camera or lens is mounted.

4.5 UAV states

For autonomous control in the test application three states are identified: *Hovering & Searching*, *Approaching* and *Facing*. The state determines how the controllers are connected to the UAV axes and which calculations are executed to investigate points of interest. A top view of the states is shown in Figure 6, and the corresponding state diagram is shown in Figure 7.

After taking off autonomously, the UAV enters the *Hovering & Searching* state. In this state the position of the UAV needs to stay fixed relative to the ground, while it rotates about its z-axis using yaw to scan the surroundings for POIs with the front camera. The drift from the hovering point is calculated by analyzing the relative translation using the bottom camera. The relative translation error is corrected by the pitch and roll axis of the UAV. The throttle is used to keep the

height constant.

Whenever a POI is located with the front camera, the UAV is set to the *Approaching* state. In this state the UAV estimates the 3D position of the POI and navigates towards it using the pitch. The yaw and throttle are used to keep the POI centered in the image. The combination of pitch and yaw tends to make the UAV drift laterally, in the direction opposite to that of the yaw. A small roll correction proportional to the amount of yaw is used to compensate this.

If the UAV is close enough to measure the orientation of the POI, the *Facing* state is entered which centers and aligns the UAV perpendicular to it. The roll is used to align the UAV, the pitch is used to keep the distance constant, and the yaw and throttle are used to keep it centered. When the POI is lost, the state is restored to *Hovering & Searching*.

The remainder of this section describes the theory and ideas behind the control of each of the states.

4.6 Hovering & Searching

During the *Hovering & Searching* state, dense optical flow using the Farnebäck algorithm [14] is used. This algorithm estimates the drift of the UAV by analyzing two consecutive frames and estimating the translation and rotation using a pixel descriptor based on polynomial expansion. A constant altitude is maintained using the ultrasonic sensor.

4.7 Approaching

When the POI is found in the *Hovering & Searching* state, the *Approaching* state is entered. During this state the following tasks need to be completed:

1. Locate the POI with sub-pixel precision using the front camera and image analysis
2. Calculate the horizontal, vertical and distance deviations of the UAV with respect to the POI
3. Control these deviations to zero using yaw, throttle and pitch respectively

For the rectangular POI, the horizontal and vertical deviations are calculated as follows. The height of the target in the real world is known. In an image taken by the UAV the rectangle has a certain pixel height. With this it can be calculated how many real-world millimeters correspond to one pixel in the image:

$$\text{Scale factor} = \frac{\text{Real rectangle height (mm)}}{\text{Pixel rectangle height (px)}} \quad (1)$$

The horizontal and vertical deviations in pixels can be found by comparing the center of the image to that of the POI in the image. Using the scale factor, these pixel deviations can be converted to deviations in millimeters.

The distance can be calculated using the equation for the magnification M in optics, using the thin lens model:

$$M = \frac{h_{\text{object}}}{h_{\text{image}}} = \frac{d_{\text{object}}}{d_{\text{image}}} \quad (2)$$

in which h_* and d_* are heights and distances in millimeters. The distance that should be calculated is d_{object} . Rewriting then gives:

$$d = d_{object} = d_{image} \frac{h_{object}}{h_{image}} = f \frac{\text{Rectangle height (mm)}}{\text{Rectangle height (px)} \times \text{Pixel size (mm/px)}} \quad (3)$$

The effective focal length f and pixel size can be deduced from the specifications of the lens and camera. This distance is controlled to the set-point of the *Approaching* state's distance controller, which is equal to the set-point of the distance controller in the *Facing* state. As soon as the set-point is reached, the UAV transitions into the *Facing* state.

4.8 Facing

The *Facing* state does the same as the *Approaching* state, plus the correction of the perspective angle between the UAV and the POI. This angle can only be calculated reliably when the UAV is close to the POI and the POI is centered in the image. A schematic top view of the situation which uses the pinhole camera model is shown in Figure 8.

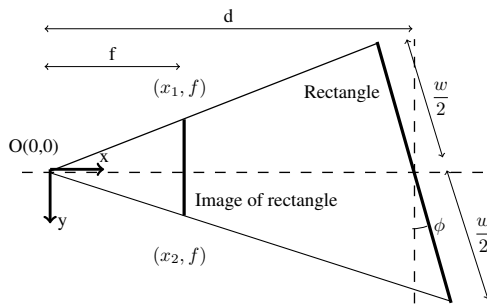


Figure 8: Top view of the camera at point O with focal length f facing the rectangle at an angle ϕ .

The camera with focal length f is located at point O . Depending on the perspective angle ϕ between the UAV and the POI, the ratio between the part of the image of the POI to the left of the center of the image (x_1) and the right (x_2) will change. The width of the real-world rectangle is given by w . From geometry it follows that ϕ is given by:

$$\tan \phi = \frac{1}{2} \left(\frac{f}{x_1} + \frac{f}{x_2} \right) \quad (4)$$

The angle is controlled to zero in order to directly face the POI, whereas the distance is controlled to a chosen set-point in front of it according to Equation 3, as explained before.

The theory described above assumes that the POI is not rotated about its horizontal axis. However, it can be extended to take this rotation into account.

5 EXPERIMENTS

A test application to test the Twirre architecture with autonomous flight has been developed. The test setting is shown in Figure 9. This section describes the tests that have been performed.



Figure 9: Experimental setting used during experiments.

5.1 Autonomy switch

The autonomy switch plays a central role in safety during the experiments. If a controller results in unstable behavior, the pilot can instantaneously take over the control to correct the movement of the UAV. The reliability of the switch is researched in a qualitative way, as well as its switching speed.

5.2 Hovering & Searching

The hovering is tested by taking off autonomously, setting the hovering set-point and letting the UAV hover autonomously at that point. The set-point of the height is at 1 meter above the ground. The hovering performance is measured by calculating the average longitudinal, lateral and height deviation during flight, along with their standard deviation and maximum value.

5.3 Approaching and Facing

During the *Approaching* and *Facing* experiments, autonomous control of every axis is tuned separately. So for the *Approaching* state, the yaw, throttle and pitch are tuned separately first. The PID gains are improved by looking at graphs of the UAV's response, showing deviations and steering inputs over time. After that, the controllers are combined, followed by fine-tuning in order to improve the combined control. The states are tested with a distance set-point of 3 meters from the POI.

6 RESULTS

The results obtained from the experiments explained above are shown in this section.

6.1 Autonomy switch

The autonomy switch was used extensively during the experiments. It was found perfectly reliable, as it worked correctly and instantly every time it was used.

Table 1: Hovering accuracy measurements.

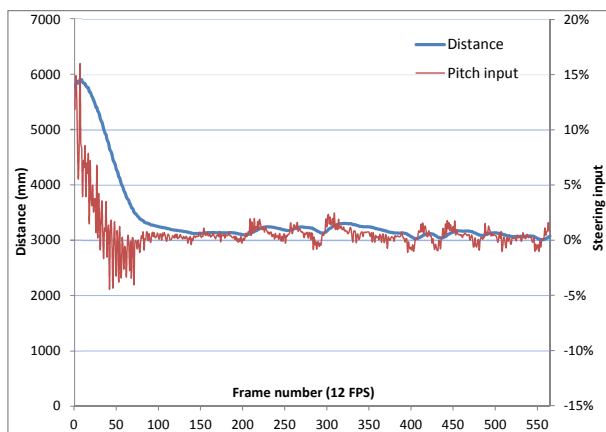
Deviation	Average (mm)	Std dev (mm)	Max (mm)
Longitudinal	-7.7	19	40
Lateral	9.2	21	44
Height	-7.4	8.6	30

6.2 Hovering & Searching

The hovering controllers result in the accuracy shown in Table 1. The numbers have been taken over nearly 600 frames at 40 FPS. It can be concluded that the hovering is accurate, with little deviations from the set-point.

6.3 Approaching and Facing

The resulting distance response during the *Approaching* state is shown in Figure 10. It shows the distance accurately converging to the set-point of 3 meters from the POI.

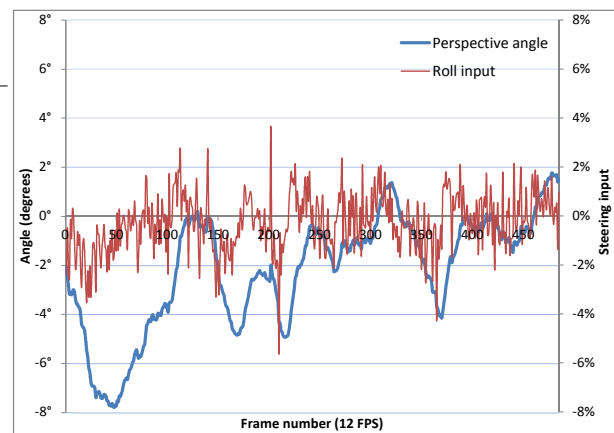
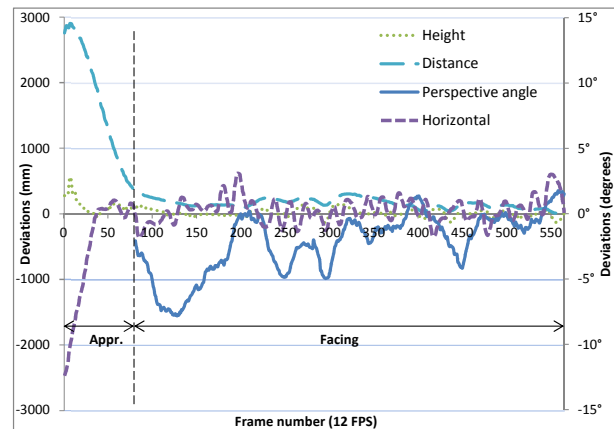
Figure 10: Distance control during the *Approaching* state.

The perspective angle deviation over time along with the roll steering inputs during the *Facing* state is shown in Figure 11. It shows the angle converging to zero.

Combining the *Approaching* and *Facing* state gives the performance shown in Figure 12. The *Approaching* state is active until frame 80 as can be seen in the perspective angle, which is in that case assumed to be zero because it cannot be measured reliably. For the state transition a distance margin of 0.4 meters around the set-point is used. As soon as the UAV enters this margin and the POI is centered, it transitions to the *Facing* state. This happens at frame 80, after which the perspective angle is no longer assumed zero. It can be seen that this angle is also controlled to zero. The graph shows accurate performance of controlling all four movements (roll, pitch, yaw and throttle) autonomously at the same time.

7 CONCLUSION

The Twirre architecture for UAVs consists of low-cost, easily interchangeable commodity hardware components. This

Figure 11: Perspective angle control during the *Facing* state.Figure 12: Deviations over time while performing the *Approaching* and *Facing* states after each other.

architecture has been used to implement a hexacopter using standard RC hobby components.

In the process of building and testing the hexacopter multiple components have been exchanged and were upgraded, while other components remained fixed. This was made easy because of the modular design of the Twirre architecture.

The autonomy switch is used to instantly switch back to manual mode without interference from the autonomous controller. Tests confirm that this is a reliable method for switching between manual and autonomous control.

A test application to demonstrate autonomous flight in a GPS-deprived environment is developed. Several experiments for autonomous flight towards a point of interest have been performed. The Twirre architecture allows for separate tuning of the high-level controllers. Experiments show that the tuned controllers show good performance in approaching and facing an object.

8 FUTURE WORK

In the test application the software for the mission logic, the sensor data processing and the high-level control system are intertwined. Future work is to design and build reusable, non application specific software components, from which autonomous applications can be built. The goal is to separate the application specific software parts from the architecture, which makes the Twirre architecture more versatile for multiple applications. Future work will focus on the software design for the high-level control. This includes extending the state machine.

The current platform will also be combined with other sensors, like GPS, 3D sensors/cameras and multiple ultrasonic sensors. This will extend the platform to a multi-sensor fusion for robust autonomous flight in indoor and outdoor environments. The bottom camera will be put on a new downward-facing gimbal in order to have accurate optical flow measurements when moving over large distances. This will make it possible to implement new autonomous states like homing and landing.

9 REFERENCES

- [1] Micropilot. *MP2x28 Family of UAV Autopilots*.
- [2] Cloud Cap Technology. *Cloud Cap Technology Piccolo SL*.
- [3] DJI. *A2 Flight Control System User Manual*, 1.14 edition, 02 2014.
- [4] Shaojie Shen, Yash Mulgaonkar, Nathan Michael, and Vijay Kumar. Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft mav. In *2014 IEEE International Conference on Robotics & Automation (ICRA)*, 05 2014.
- [5] G CHE de Croon, HW Ho, C De Wagter, E Van Kampen, B Remes, and QP Chu. Optic-flow based slope estimation for autonomous landing. *International Journal of Micro Air Vehicles*, 5(4):287–298, 2013.
- [6] Laurie N Bose and Arthur G Richards. Mav belief space planning in 3d environments with visual bearing observations. In *International Micro Air Vehicle Conference and Flight Competition (IMAV2013)*, 2013.
- [7] Bart Remes, Dino Hensen, Freek van Tienen, Christophe de Wagter, Erik van der Horst, and Guido de Croon. Paparazzi: how to make a swarm of parrot ar drones fly autonomously based on gps. In *International Micro Air Vehicle Conference and Flight Competition (IMAV2013)*, Toulouse, France, 2013.
- [8] Andrew Nolan, Daniel Serrano, Aura Hernandez Sabaté, Daniel Ponsa Mussarra, and Antonio M López Pena. Obstacle mapping module for quadrotors on outdoor search and rescue operations. In *International Micro Air Vehicle Conference and Flight Competition (IMAV2013)*, Toulouse, France, 2013.
- [9] Balazs Gati. Open source autopilot for academic research the paparazzi system. In *proceeding of the American Control Conference 2013*, 2013.
- [10] Jan Bolting, Francois Defay, and Jean-Marc Moschetta. Differential gps for small uas using consumer-grade single-frequency receivers. In *International Micro Air Vehicle Conference and Flight Competition (IMAV2013)*, 2013.
- [11] Jose Luis Sanchez-Lopez, Jesus Pestana, Paloma de la Puente, and Pascual Campoy. Visual quadrotor swarm for imav 2013 indoor competition. In *ROBOT2013: First Iberian Robotics Conference*, pages 55–63. Springer International Publishing, 2014.
- [12] Kirill E. Shilov, Vladimir V. Afanasyev, and Pavel A. Samsonov. Vision-based navigation solution for autonomous indoor obstacle avoidance flight. In *International Micro Air Vehicle Conference and Flight Competition (IMAV2013)*, Toulouse, France, 2013.
- [13] Ming Liu, Cédric Pradalier, François Pomerleau, and Roland Siegwart. Scale-only visual homing from an omnidirectional camera. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3944–3949. IEEE, 2012.
- [14] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, pages 363–370. Springer, 2003.
- [15] G. Klein; D. Murray. Parallel tracking and mapping for small ar workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, 2007.
- [16] Bloesch M., Weiss S., D. Scaramuzza, and Siegwart R. Vision based mav navigation in unknown and unstructured environments. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2010.
- [17] Benjamin Ranft, Jean-Luc Dugelay, and Ludovic Apvrille. 3d perception for autonomous navigation of a low-cost mav using minimal landmarks. In *International Micro Air Vehicle Conference and Flight Competition (IMAV2013)*, Toulouse, France, 2013.

Pitch and Yaw Control of Tailless Flapping Wing MAVs by Implementing Wing Root Angle Deflection

CHAN Woei Leong,* NGUYEN Quoc Viet, and Marco DEBIASI
National University of Singapore

ABSTRACT

Study was conducted on the effect of wing root angle deflection to pave the way for the implementation of tailless flight control on a hovering flapping wing platform. Hypotheses of how the wing root deflection in the sagittal direction could help generate pitching and yawing moments are presented and investigated. The investigation was completed using a four bar linkage crank-rocker flapping mechanism. The flapping frequencies are from 13Hz to 18Hz, and the sagittal deflection angle ranged between -6.7° to 7° . Study shows that tailless pitch and yaw control is feasible but the nonlinearities could lead to huge challenge on implementation. The deflection angle also compromises lift.

1 INTRODUCTION

Flapping wing micro air vehicle (FW-MAV) has been around for quite some time. Earlier flapping wing MAVs such as the DelFly [1] employs tail control. Pitch, and yaw moments are generated by conventional control surfaces like the fixed wing aircraft. In the latest edition of the DelFly, the DelFly Explorer [2] has a pair of ailerons for roll control. Due to biomimicry reason, tailless control has gained a lot of interests recently. Unlike birds, insects fly without using tail; the only way to manoeuvre is by changing the wing kinematics. Dickinson [3] argued that the rotational force plays an important part in insect flight control and the insects regulates rotational phase for flight manoeuvres. Dickinson also explained explicitly how the insects perform flight control about the three rotational axes [4]: mean stroke position is altered to generate pitch moment; yaw is achieved by differential adjustment of stroke amplitude; roll is achieved by differential inclining of the stroke planes of the wings.

The first successful implementation of tailless control on a standalone artificial platform was demonstrated by Keennon [5] on the Nano Hummingbird. The Nano Hummingbird uses a combined wing twist and rotation modulation to achieve flight control. The other iconic standalone flapping wing platforms that utilize tailless control are the Techjets Dragonfly [6], and the FESTO BionicOpter [7]. The smallest, but not standalone tailless flapping wing platform is the

Robobee [8, 9]. Other than that, Phan *et al.* [10] designed a flapping mechanism that can generate pitching moment by changing the stroke angle range, a mechanism similar to the real insect, Oppenheimer *et al.* [11] studied the utilization of split-cycle constant-period frequency modulation for flapping wing flight control, and Hale *et al.* [12] tried to do so by varying wing membrane stiffness.

The Temasek Laboratories of the National University of Singapore has started developing flapping wing MAVs since two years ago. Insect inspired prototypes are built and flight tested. A four-winged and a two-winged hovering flapping wing platform with tail control was developed. However, the flight quality of the two-winged platform is deficient due to unsound flight control and stability. The present work is focused on studying the effect of deflecting the wing root in the sagittal direction (forward and backward). We hypothesized that the deflection will result in changes of pitching and yawing moments. Understanding the effect of moment changes will help pave the way of utilizing the effect for flight control in future tailless flapping wing platform. Section 2 of this paper explains the hypotheses of pitch and yaw moment generation by deflecting the wing root. The flapping mechanism and the wing, and the experimental setup are being elaborated in Section 3 and 4 respectively. Force and moment measurements are presented and discussed in Section 5. Finally, the conclusions end the paper.

2 HYPOTHESES

Figure 1 illustrates the right wing deformation during a downstroke at positive deflection angle. Deflection to the front (ventral) is defined as positive. The angle is parallel to the sagittal plane, thus called sagittal deflection angle ($\delta_{s,r}$).

2.1 Pitching Moment

It is defined in this paper that a ventral-stroke is when the leading edges of the wings are at the ventral side of the flapping wing platform. The end of downstroke and the beginning of upstroke (supination) occurs during the ventral-stroke. Similarly, the dorsal-stroke is when the leading edges of the wings are at the dorsal side.

At a positive sagittal deflection angle, the wing is slacker at the ventral-stroke. We hypothesized that a slacker wing would allow more wing rotation, and more lift to be generated. Likewise, a stretched wing at the dorsal-stroke discourages wing rotation and produces less lift. The lift difference results in an unbalanced pitching moment. A positive sagittal

*Email address(es): tslcwl@nus.edu.sg

deflection angle on either right or left wing will result in a pitch up moment as illustrated in Figure 1.

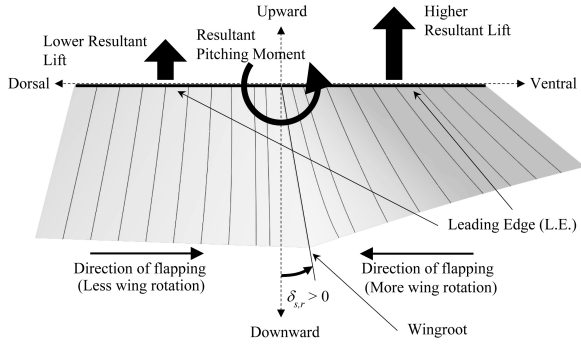


Figure 1: Pitching moment generated due to wing root deflection viewed from right side.

2.2 Yawing Moment

At a positive sagittal deflection angle, the right wing is rotated more during the upstroke, thus reducing drag. The same phenomenon happens when the left wing has a negative sagittal deflection angle during downstroke. We hypothesized that the drag difference between the down and upstrokes creates an unbalanced yawing moment. A positive and negative sagittal deflection angles on the right and left wings respectively will result in a positive yawing moment as illustrated in Figure 2.

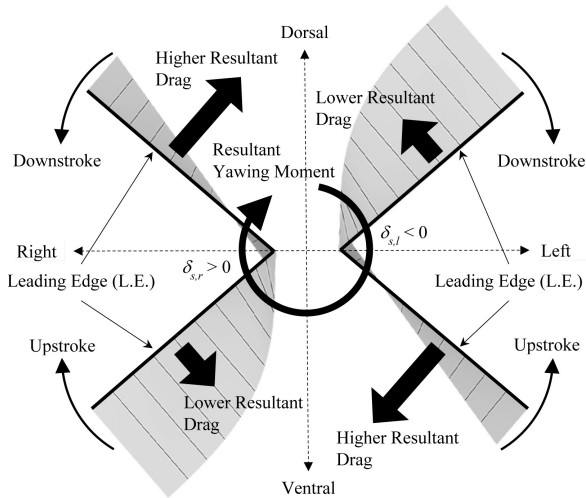


Figure 2: Yawing moment generated due to wing root deflection.

3 FLAPPING MECHANISM AND WING

3.1 Flapping Mechanism

The flapping mechanism being used in the experiment is a four bar linkage crank-rocker system as shown in Figure 3. The gearbox structural components were cut out of a 1mm thickness glass epoxy panel using a CNC milling machine.

The gears are off-the-shelf Nylon gears and brass pinion with a total gear ratio of 16:1. OverSky HP03S 16600kV brushless motor was used to drive the mechanism. The stroke angle (ϕ) range is $\pm 64^\circ$. Critical dimensions of the gearbox are $l_0 = 12.6\text{mm}$, $l_1 = 4.5\text{mm}$, $l_2 = 12.5\text{mm}$, and $l_3 = 5.0\text{mm}$. Doman *et al.* [13] has derived mathematical models for a four bar linkage mechanism. The models are presented here as Equation 1 to 4. Equation 5 represents the relation of the rocker output angle (ϕ_l) to the stroke angle (ϕ) of the flapping mechanism presented in this paper.

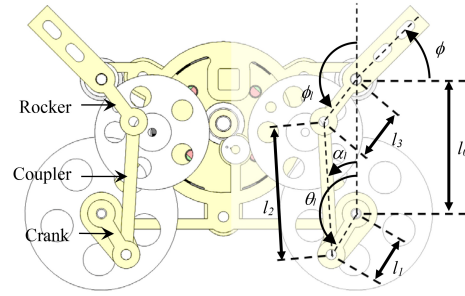


Figure 3: Four bar linkage crank-rocker flapping mechanism.

$$\alpha_l = \arctan 2 \left(\frac{l_3 \sin \phi_l - l_1 \sin \theta_l}{l_0 + l_3 \cos \phi_l - l_1 \cos \theta_l} \right) \quad (1)$$

$$\phi_l = 2 \cdot \arctan 2 \left(\frac{-K_1 \pm \sqrt{K_1^2 - K_3^2 + K_2^2}}{K_3 - K_2} \right) \quad (2)$$

where $K_1 = -2l_1 l_3 \sin \theta_l$

$$K_2 = 2l_3(l_0 - l_1 \cos \theta_l)$$

$$K_3 = l_0^2 + l_1^2 - l_2^2 + l_3^2 - 2l_0 l_1 \cos \theta_l$$

$$\dot{\alpha}_l = \frac{l_1 \sin(\phi_l - \theta_l)}{l_2 \sin(\alpha_l - \phi_l)} \dot{\theta}_l \quad (3)$$

$$\dot{\phi}_l = \frac{l_1 \sin(\alpha_l - \theta_l)}{l_3 \sin(\alpha_l - \phi_l)} \dot{\theta}_l \quad (4)$$

$$\phi = \phi_l - 97.63^\circ \quad (5)$$

Assumed that the crank angle (θ_l) varies at constant angular rate, Figure 4 shows the ideal stroke angle within one flapping cycle under no load condition calculated using Equation 5. It shows that the downstroke and upstroke are not symmetrical. The angular rate of upstroke is generally constant, but the downstroke starts at a lower angular rate and accelerates throughout the stroke. The duration of upstroke is fractionally longer than the downstroke.

Assumed there is no power loss throughout the crank-rocker, one is able to calculate the ratio of output torque to input torque using Equation 6. Figure 5 shows the torque ratio throughout the strokes. Apparently the torque ratio at

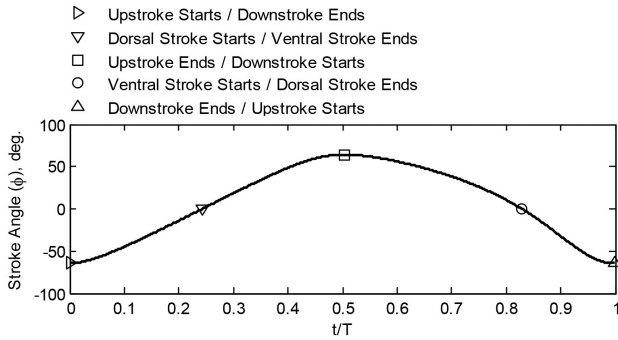


Figure 4: Ideal stroke angle without load.

the beginning of downstroke is significantly higher and reduces throughout the half stroke, while torque ratio variation of upstroke is relatively flat. It suggests that under loaded condition, the stroke angle changes rapidly at the beginning of downstroke, possibly produce higher lift as compared to the upstroke. The average torque ratio of downstroke is also higher than the upstroke. Since the stroke angle ranges of downstroke and upstroke are symmetrical, it suggests that under loaded condition, the duration of downstroke is shorter. Similarly, the duration of dorsal-stroke is supposedly shorter.

$$\frac{\tau_0}{\tau_i} = \frac{l_3 \sin(\alpha_l - \phi_l)}{l_1 \sin(\alpha_l - \theta_l)} \quad (6)$$

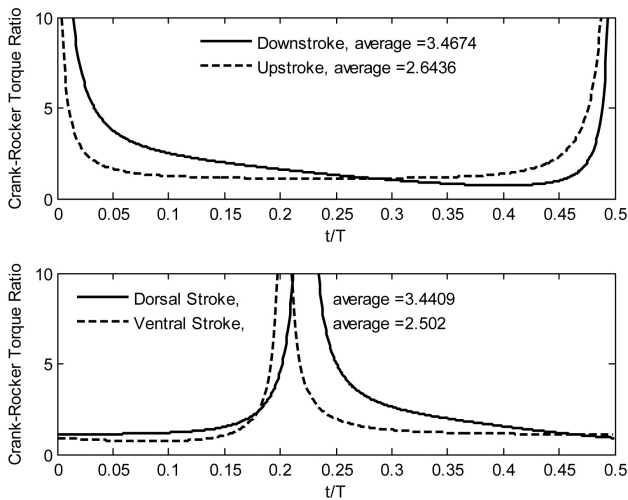


Figure 5: Flapping mechanism crank-rocker torque ratio.

3.2 Wing

Figure 6 shows a drawing of the wing used for the experiment. The wing is in rectangular shape of which the half wingspan is 90mm and the chord length is 50mm. At the wing root, there is a 10° of slack angle. The wing root and leading edge are perpendicular to each other after installation onto the flapping mechanism. Figure 7 and 8 show the wing being attached on the experimental setup. As shown in the

figures, the wing root slack angle allows the wing to have limited rotation about the leading edge and encourages wing rotation during flapping.

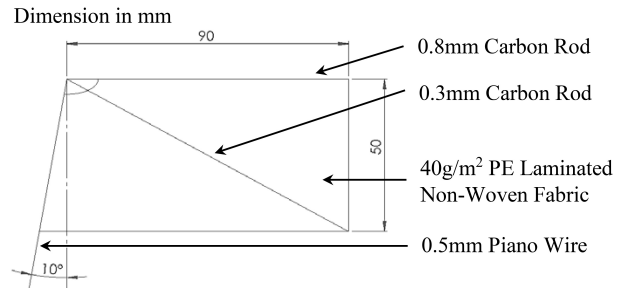


Figure 6: Wing dimension and materials.

The wing membrane is 40g/m² polyethylene (PE) laminated non-woven fabric. Non-woven fabric was chosen instead of the more frequently used Mylar because it can easily adhere to adhesives and it produces lesser acoustic signal. Non-woven fabric is porous, and porosity reduces lift. Due to that, it is required to have a layer of PE lamination. As illustrated in Figure 6, the leading edge of the wing is a 0.8mm carbon rod, the wing root is 0.5mm piano wire, and the wing stiffener is made out of 0.3mm carbon rod. The total weight of one wing is 0.6g.

4 EXPERIMENTAL SETUP

A test fixture as shown in Figure 7 is specifically designed and fabricated for the experiment. The test fixture accommodates the flapping mechanism. The wing holder holds the leading edge of the wing and the bottom end of the wing root is secured on a rotatable arm. The fixture allows adjustment of the wing sagittal deflection angle by securing the rotatable arm on the fixture through different threaded holes. It also allows the adjustment of coronal deflection, but the effect of coronal deflection will not be discussed in this paper.

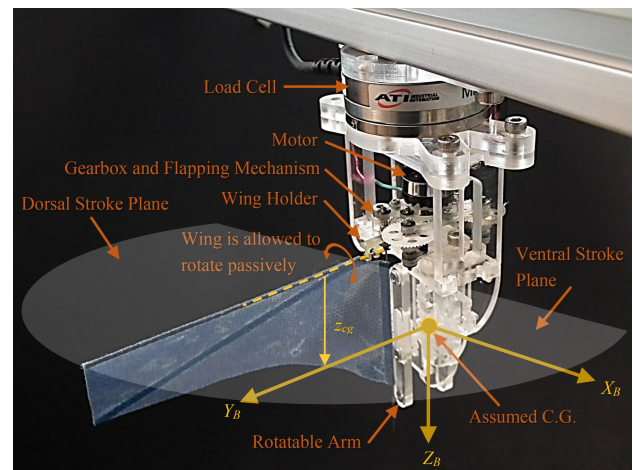


Figure 7: Experimental setup.

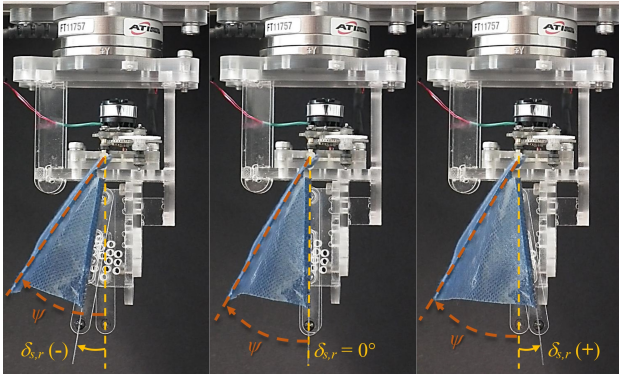


Figure 8: Side view of experimental setup showing the sagittal deflection angle of the right wing.

A load cell (ATI Mini40) is installed on top of the test fixture, and the load cell is fixed on a beam. The load cell measures 3-axis forces and 3-axis moments at a sampling rate of 5000Hz. The measured data was then transformed to the body-fixed coordinate system about the assumed centre of gravity (CG) of the eventual flying platform using Equation 7. The assumed CG is the intended centre point of the flying platform rotational motion. The distance of the assumed CG from the leading edge is being designated as z_{cg} . The origin of the body-fixed coordinate system is the assumed CG, X_B -axis points forward, Y_B -axis points to the right side, and Z_B -axis points downward. F_x , F_y , F_z , and M_x , M_y , M_z , are forces and moments about the X_B , Y_B , and Z_B -axis respectively.

$$M_{y,z_{cg}} = M_{y,z_{cg}=0} - F_x z_{cg} \quad (7)$$

where $M_{y,z_{cg}=0}$ is the pitching moment at $z_{cg} = 0$

Only the right wing is used for the experiment in order to better study the effect of the deflection. If needed, the effect of two wings can be easily deduced from the single wing data by assuming that the two wings are identical using $F_{x,l} = F_{x,r}$, $F_{y,l} = -F_{y,r}$, $F_{z,l} = F_{z,r}$, $M_{x,l} = -M_{x,r}$, $M_{y,l} = M_{y,r}$, and $M_{z,l} = -M_{z,r}$, where subscripted r and l represents right and left wing respectively. The data presented in this paper is of single wing (right wing) measurement unless mentioned otherwise.

The wing is allowed to rotate passively about the wing holder. Figure 8 shows the right view of the experimental setup with sagittal deflection angle of the right wing. From the figure, one can conceive that the wing will be slacker throughout the ventral stroke at a positive sagittal deflection angle as described earlier. A slacker wing allows more rotational angle (ψ) during the flapping motion. Similarly, a negative sagittal deflection angle encourages larger wing rotation at the dorsal-stroke.

Tests were conducted from 13 to 18Hz flapping frequency, the sagittal deflection angle ranged between -6.7° to

7° , and without incoming flow to simulate hovering condition. Given the kinematics viscosity of $1.57 \times 10^{-5} m^2 s^{-1}$, the equivalent Reynolds number as defined by Shyy *et al.* [14] is in between 16637 to 23036. The force and moment data presented in the paper are not pure aerodynamic force, they are the sum of aerodynamic and inertia forces, thus the data are not made dimensionless.

5 RESULTS AND DISCUSSION

5.1 Forward Force (F_x)

Figure 9 shows the variation of forward force (F_x) in one flapping cycle and the corresponding stroke angle. The stroke angle was captured from the images taken by a PCO 1600 sensitive camera. The horizontal axis of the figure is time being normalized to the flapping cycle period. The force data is an average of 18 repeated flapping cycles. The error bar shows that the force measurement is repeatable. Likewise, all other data of force and moment variations in a flapping cycle shown in the paper are the result of averaging the measurement of multiple cycles at their corresponding flapping frequency and sagittal deflection angle. Forward force is dominated by inertia force and as shown in the figure, it reaches the minimum at the beginning of downstroke (supination). All data of force and moment variations in a flapping cycle shown in the paper are arranged such that they start at the point of minimum forward force which presumably the point of supination. The wing kinematics wasn't monitored for all cases due to hardware limitation.

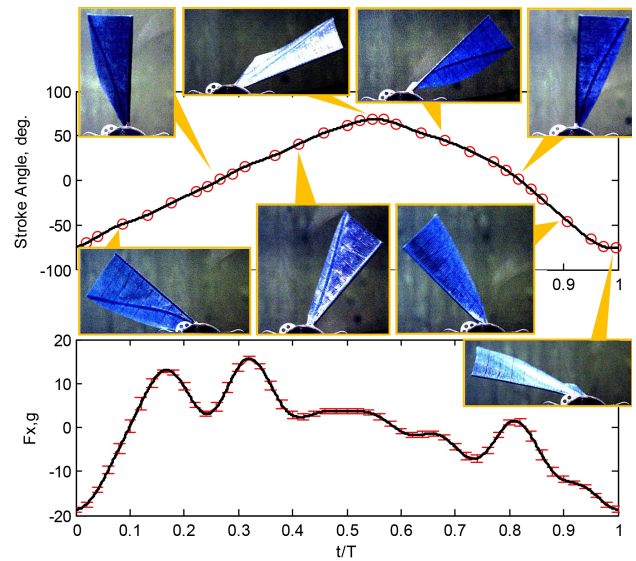


Figure 9: Stroke angle and corresponding forward force (wings as seen from the top).

Figure 10 shows the forward force variations in a flapping cycle at various flapping frequencies and sagittal deflection angles. The data starts at the beginning of upstroke, which corresponds to the minimum. It follows by two posi-

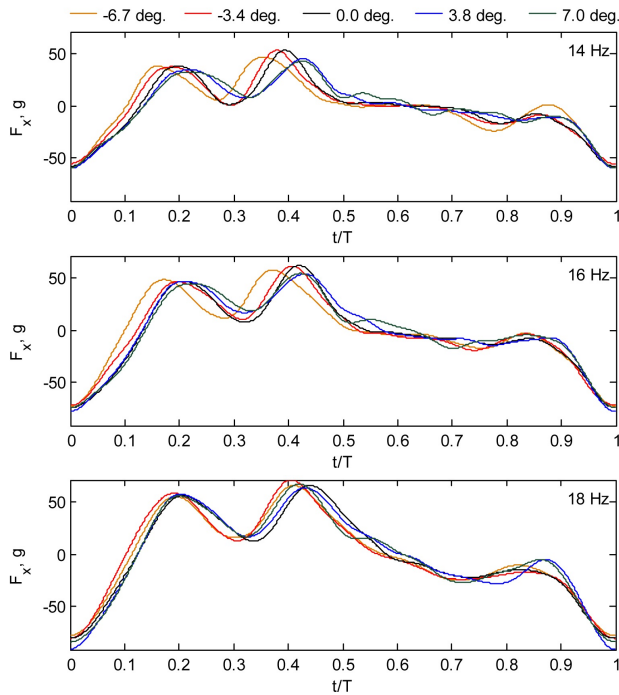


Figure 10: Forward force in one flapping cycle.

tive peaks. The figure shows that the more positive the sagittal deflection angle is, the later the peaks tend to occur. From Figure 9, the peaks occur during the first half of the downstroke when the wing is still in the ventral side. The delay could be due to the fact that the more positive the sagittal deflection angle is, the slacker the wing is at the ventral side

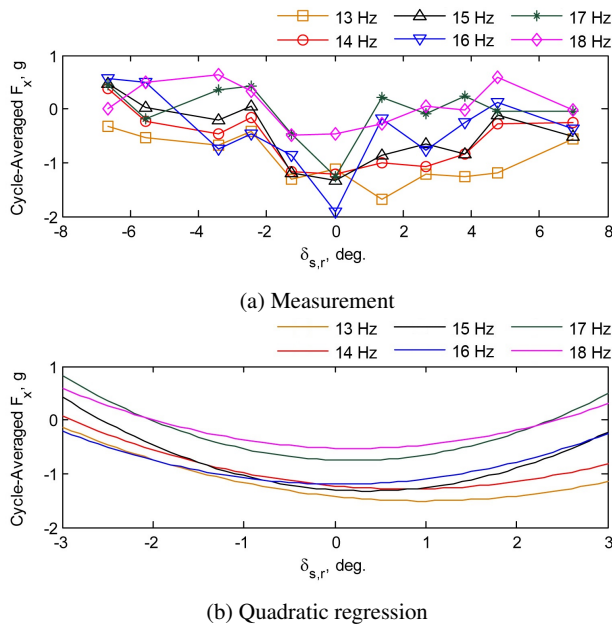


Figure 11: Cycle-averaged forward force.

and thus it takes longer to be fully rotated. However, as the flapping frequency increases, this phenomenon subsides.

Figure 11 shows the cycle-averaged forward force ($\bar{F}_x$). At zero sagittal deflection angle, the cycle-averaged forward force is negative, but the magnitude is less than 2g. As the sagittal deflection angle moves away towards either the positive or negative direction, the magnitude of cycle-averaged forward force tends to decrease. The cycle-averaged forward force is in the range of -2g to 1g, it should not have significant effect on the translational motion, but it has significant impact on the pitching moment as discussed later. Figure 11(b) shows the result of quadratic regression of the cycle-averaged forward force within a smaller range of the sagittal deflection angle. Apparently, the cycle-averaged forward force is at its maximum magnitude at 0° to 1° sagittal deflection angle.

5.2 Lift Force (L)

The lift force ($L = -F_z$) is being presented in Figure 12. There are peaks of lift force right after the start and at the middle of the upstroke. We will refer to the peaks as the first and second peak respectively. The first peak reduces and the second peak increases as the sagittal deflection angle increases. The first peak occurs after supination, thus its reduction with sagittal deflection angle suggests that the slacker wing do not guarantee higher lift generation after all. Also observed is the delay of the second peak as the sagittal deflection angle increases. Negative peaks are also present right before the positive peaks, and the magnitude of the negative peaks increases as the corresponding positive peaks increases.

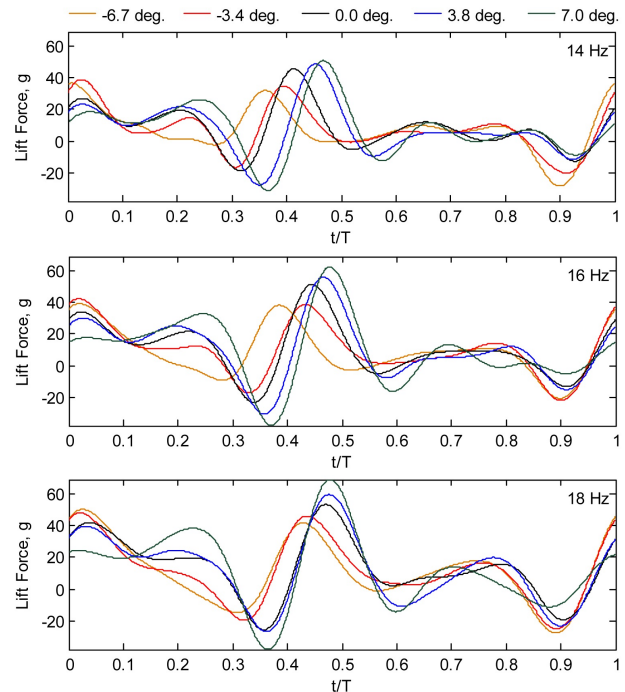


Figure 12: Lift force in one flapping cycle.

As the flapping frequency increases, the magnitude of the positive peaks increases, but the magnitudes of the negative peaks do not increase significantly, resulting in an increase of total lift as shown in Figure 13. Figure 13 shows that the magnitudes of the cycle-averaged lift ($\bar{L} = -\bar{F}_z$) are at the maxima when the wing root is not deflected, indicating that tailless control compromises lift.

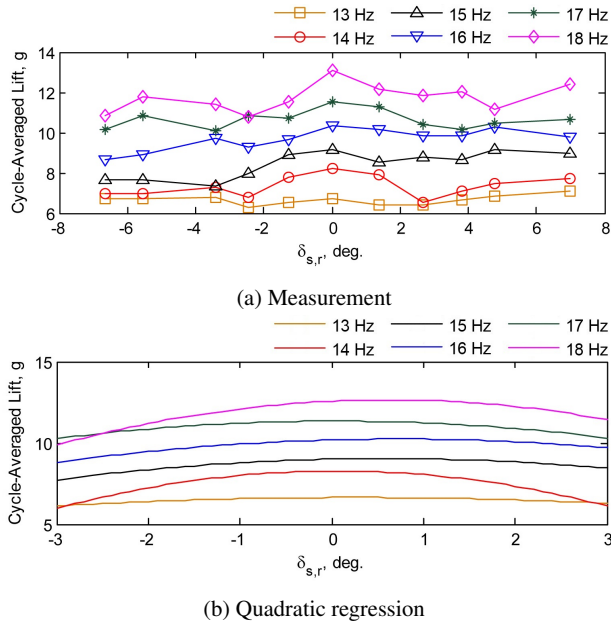


Figure 13: Cycle-averaged lift.

5.3 Pitching Moment (M_y)

Figure 14 clearly shows the pitching moment is more closely correlated to the forward force than the lift. Evidently, the pitching moment hypothesis is over simplified. Figure 15 shows the pitching moment variation at 18Hz at six difference CG locations, namely half chord distance above the stroke plane ($z_{cg}=-25\text{mm}$), the stroke plane ($z_{cg}=0\text{mm}$), half chord below ($z_{cg}=25\text{mm}$), and one chord below ($z_{cg}=50\text{mm}$) the stroke plane respectively. As shown in the figure, the further away the CG from the half chord below stroke plane ($z_{cg}=25\text{mm}$) in either directions, the pitching moment become more correlated to the forward force. This is due to the distance of the CG and the acting point of forward force. The distance is the moment arm, and the pitching moment is dominated by the forward force when the length of moment arm is significant. One must not confused this forward force acting point with the centre of pressure because the forces measured are the sum of aerodynamic force and inertia force. From the observation, it can be safely inferred that the average vertical coordinate of the forward force acting point is at the half chord below the stroke plane ($z_{cg}=25\text{mm}$).

As shown in Figure 16, as the CG being shifted downward, the cycle-averaged pitching moments ($\bar{M}_y$) tend to have double troughs patterns with the centre peaks at zero

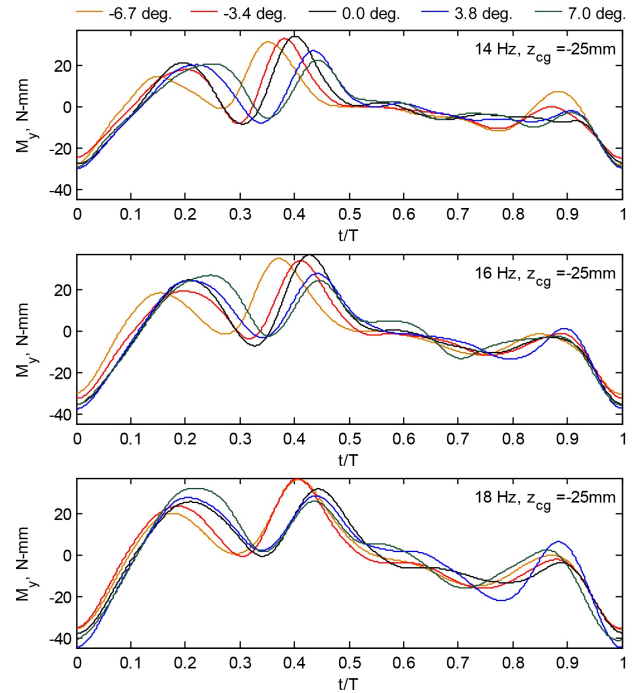


Figure 14: Pitching moment in one flapping cycle.

sagittal deflection angle. The magnitude also tend to increase and eventually the cycle-averaged pitching moments of the lower flapping frequencies (13Hz to 16Hz) only distribute in the positive region and pitch control in these frequencies are deemed infeasible.

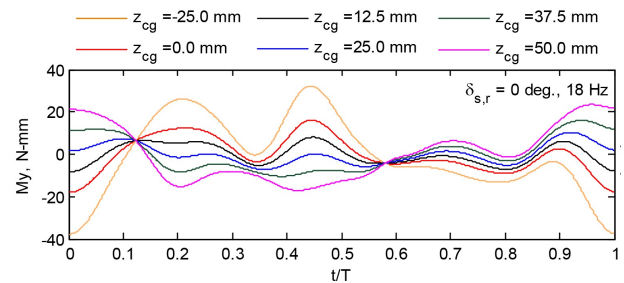


Figure 15: Pitching moment about various CG locations in one flapping cycle ($\delta_{s,r} = 0^\circ$ at 18Hz).

In summary, in order to have better pitch control characteristics and flight stability, the option is to have the CG located at half chord and flap at higher frequency. Flapping at higher frequency ensures that the cycle-averaged pitching moment is being distributed in both positive and negative regions. However, the double troughs nonlinearity will still bring about huge challenge in pitch control implementation.

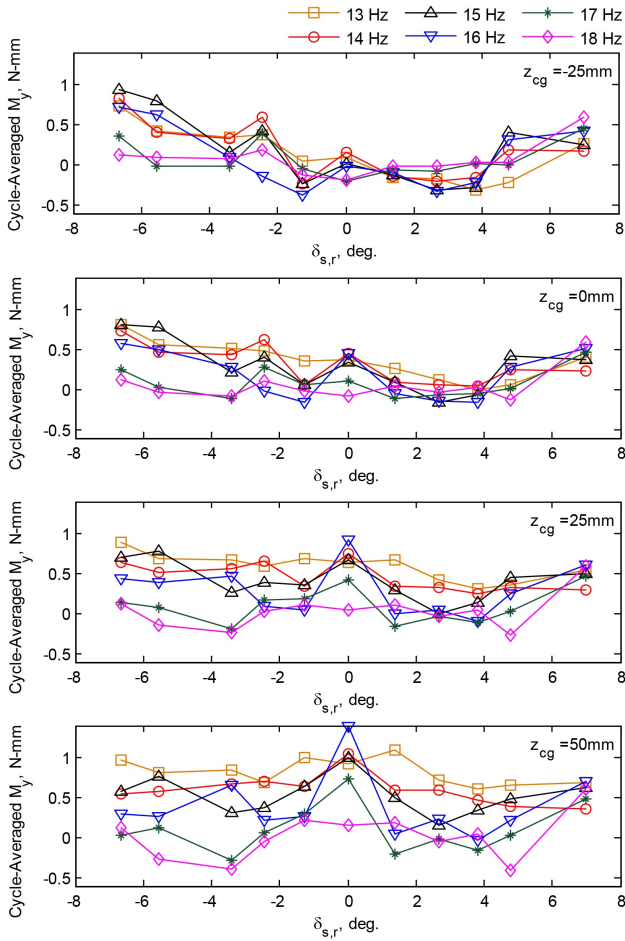


Figure 16: Cycle-averaged pitching moment.

5.4 Yawing Moment (M_z)

Assumed that the flying platform is symmetrical about the X_B - Z_B -plane, then the yawing moment is not affected by the shifting of CG along Z_B -axis. The two contributors to the yawing moment are the forward and sideward force. Figure 17 shows the variation of the yawing moment with a flapping cycle at different sagittal deflection angles for three different flapping frequencies. Similar to the pitching moment hypothesis, the yawing moment hypothesis seems to be over simplified. Instead of generating positive and negative yawing moments throughout the whole downstroke and upstroke respectively, positive and negative yawing moments are generated roughly during the dorsal-stroke and the ventral-stroke respectively. Yawing moment due to the motor rotation is assumed to be negligible because the moment is so small that the load cell is unable to measure it.

The cycle-averaged yawing moments are presented in Figure 18. In general, the yawing moments have positive slope in the negative sagittal deflection angle region and a milder negative slope in the positive sagittal deflection angle region. The cycle-averaged yawing moment is almost always

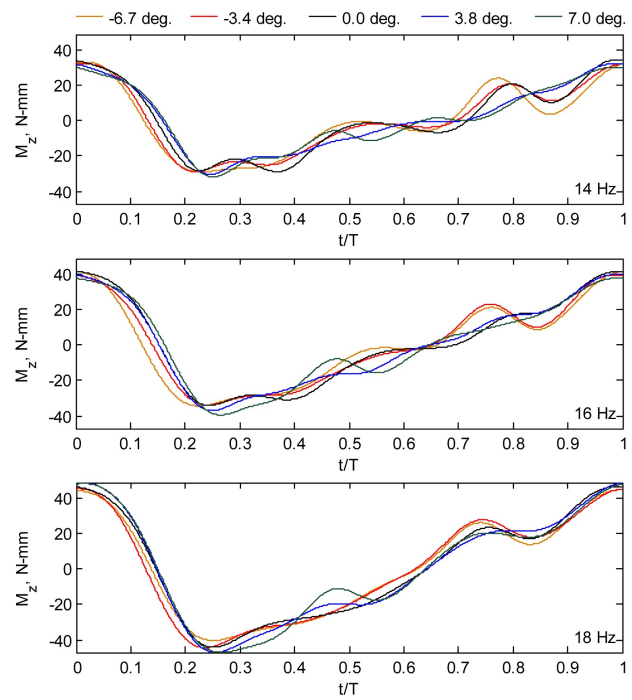


Figure 17: Yawing moment in one flapping cycle.

positive, contrary to what was hypothesized. However, yawing control is still implementable if the effect of the left wing is taken into account.

Figure 19 shows how the total cycle-averaged yawing moment ($M_{z,T}$) at 16Hz flapping frequency is being shifted vertically if an identical left wing is being deflected. The data

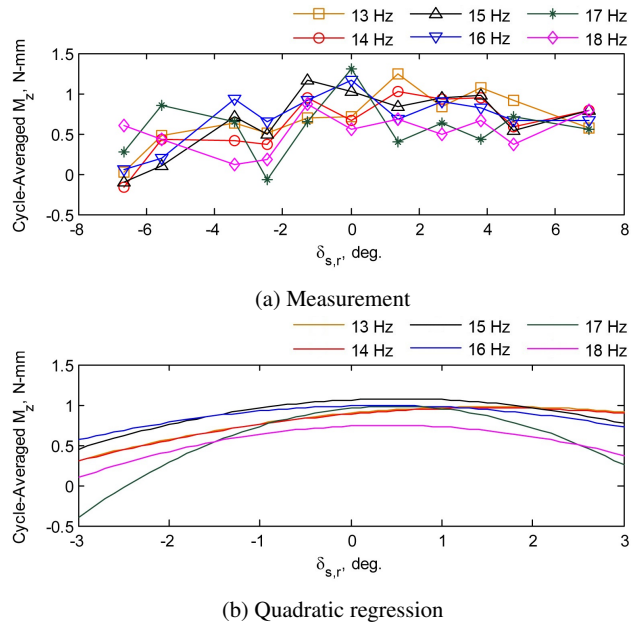


Figure 18: Cycle-averaged yawing moment.

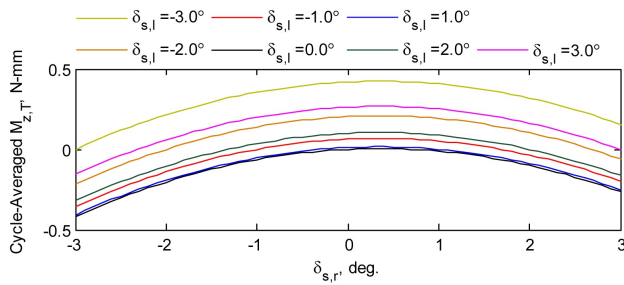


Figure 19: Total yawing moment with difference sagittal deflection angles at both wings (16Hz).

is deduced from the quadratic regression of the single wing measurement. By using different combinations of the left and right wing sagittal deflection angles, positive and negative yawing moments can be generated. Thus yawing control is possible albeit the challenge due to the nonlinearity.

6 CONCLUSION

The effect of deflecting the wing root in the sagittal direction and the possibility of using the effect for pitch and yaw control are investigated. The flapping mechanism is a four bar linkage crank-rocker system which has unsymmetrical stroke angles and torque output. The hypotheses are proven to be over simplified. A more thorough study must be conducted in order to fully understand the physical phenomena when the sagittal deflection angle is applied, such to better capitalize the effect in future designs.

It is discovered that the best CG location is at the half chord distance below the stroke plane. It is possible to implement the sagittal deflection angle for flight control, but it is a huge challenge due to the nonlinearities. Furthermore, pitch and yaw are coupled, and it is quite a challenge to find an equilibrium point where by the total sums of the moments and forces are zeroes. Apart from the pitching and yawing moments, sagittal deflection angle delays the force peaks. This effect subsides as the flapping frequency increases. Lastly, the deflection also compromises lift.

Future study could be including designing a symmetrical stroke mechanism such that the forces and moments are at equilibrium when the wings are not deflected, and conducting studies with other wing materials and configurations at higher flapping frequencies. Wing kinematics capture and flow visualization are also essential in order to have better understanding on the force and moment variations. Higher resolution on the deflection angle is also very important to capture the cycle-averaged force and moment variations accurately. Study will also be conducted to investigate the effect of wing root deflection in the coronal direction.

ACKNOWLEDGEMENTS

Authors appreciate the financial support from Future System and Technology Directorate of Singapore (Programme No. PA 9011102539).

REFERENCES

- [1] M. Groen, B. Bruggeman, B. Remes, R. Ruijsink, B. van Oudheusden, and H. Bijl. Improving flight performance of the flapping wing MAV Delfly II. In *International Micro Air Vehicle conference and competitions (IMAV)*, 2010.
- [2] TU Delft's Delfly Explorer, available at <http://www.delfly.nl/explorer.html>
- [3] M.H. Dickinson, F. Lehmann, and S.P. Sane. Wing rotation and the aerodynamic basis of insect flight. *Science*, 284 (5422): 1954-1960, 1999.
- [4] W.B. Dickson, A.D. Straw, C. Poelma, and M.H. Dickinson. An integrative model of insect flight control. In *AIAA Aerospace Sciences Meeting and Exhibit*, 2006.
- [5] M. Keennon, K. Klingebiel, H. Won, and A. Andriukov. Development of the Nano Hummingbird: a tailless flapping wing micro air vehicle. In *50th AIAA Aerospace Sciences Meeting*, 2012.
- [6] Techjet's Dragonfly, available at <http://www.techjet.com/>
- [7] Festo's BionicOpter, available at <http://www.festo.com/>
- [8] R.J. Wood. Design, fabrication, and analysis of a 3 DOF, 3cm flapping-wing MAV. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.
- [9] K.Y. Ma, P. Chirarattananon, S.B. Fuller, R.J. Wood. Controlled flight of a biologically inspired, insect-scale robot. *Science*, 340 (6132): 603-607, 2013.
- [10] V.H. Phan, T.Q. Truong, and H.C. Park. Pitching moment generation for longitudinal attitude control of insect-mimicking flapping wing system. In *10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2013.
- [11] M.W. Oppenheimer, D.O. Sigthorsson, I.E. Weintraub, D.B. Doman, and B. Perseghetti. Wing velocity control system for testing body motion control methods for flapping wing MAVs. In *51st AIAA Aerospace Science Meeting*, 2013.
- [12] L. Hale, M. Patil, and C. Woolsey. Control of flapping wing MAV using variable stiffness membrane wings. In *51st American Control Conference*, 2012.
- [13] D.B. Doman, C.P. Tang, and S. Regisford. Modeling interactions between flexible flapping wing spars, mechanisms, and drive motors. *Journal of Guidance, Control, and Dynamics*, 34(5): 1457-1473, 2011.
- [14] W. Shyy, Y. Lian, J. Tang, D. Viieru, and H. Liu. *Aerodynamics of low Reynolds number flyer*. Cambridge University Press, New York, 2008.

Design Analysis, Modelling and Experimental Validation of a Bird-like Flapping-Wing Flying Robot

Aneesh N. Chand*, Michihiro Kawanishi, and Tatsuo Narikiyo
Control Systems Lab, Toyota Technological Institute, Japan

ABSTRACT

The design of a relatively new genre of aerial robots — a full-scale, bird-like flapping wing flying robot — is analyzed, modelled and validated through real experiments in this paper. Quaternions are used instead of Euler angles in the dynamical model to represent the flying robot orientation in 3D space in order to realize smoother rotational manoeuvres using spherical linear interpolation. Effect of the control surfaces on the rotational behaviour of the flying robot is used to deduce the moments induced. The moments are then used to determine the angular accelerations, rates and orientation of the robot body in 3D space. Aerodynamic forces acting are then used to model the translational motion of the robot. For this, we propose practical methods of estimating the lift and propulsion generated by the flapping wings of the flying robot. The robot model is first simulated in a virtual environment to realize basic yawing and pitching manoeuvres and real experiments are conducted subsequently. Simulated motion corroborates with the real sensor data and gives an insight into the type of future controllers that ought to be designed.

1 INTRODUCTION

The last decade has seen concerted research and effort in the field of micro aerial vehicles (MAVs) [1]. In Japan, the need for robust and autonomous MAVs came to light in the aftermath of the Fukushima nuclear disaster. In North America, MAVs are now being touted for use in commercial delivery. The penetration of MAVs in civil applications in Europe in the coming years is expected to be so tremendous that the even the European Commission (EC) has taken a proactive stance and formulated a strategy to deal with it [2].

In the beginning, much of the related research used mainly small-sized fixed and rotary winged aerial robots [3], [4] i.e. airplanes and quadcopters. Focus then shifted to the realization of flapping-wing MAVs with fundamental design-oriented research done in [5], [6], [7]. Current work on flapping wing MAVs has largely focused on small and micro-sized aerial robots intended for indoor environments e.g. the



Figure 1: The SlowHawk2 flapping wing flying robot with articulated wings.

DelFly [8]. Thus, there is much scope to investigate in the realization of large bird-like, flapping wing MAVs capable of strong performance in outdoor flight.

The DelFly series of ornithopters [9], [10] are small-sized flapping-wing robots that could autonomously fly using only onboard power. Some models developed are perhaps the only flapping-wing flying robots to have been endowed with autonomous navigation using on-board vision control [11]. The Harvard Microrobotic Fly [12] was a biologically inspired micro aerial vehicle with flapping wings. Driven by a piezoelectric actuator, it could take off vertically while being operated through a tethered wire. Its successor, the RoboBee [13], could perform basic rolling and pitching movements while tethered. Both robots were experimental prototypes and the RoboBee is still under development. The micromechanical flying insect of [14] was a similar project as the Harvard Microrobotic Fly. The design and mathematical model of the robot was based on insect flight aerodynamics while also using non-articulated wings.

This work is discernible from related works in a number of ways. Compared to [9]-[14], this work uses a full-scale, bird-like flapping wing flying robot capable of strong outdoor flight whose flying mechanics and control surfaces are different. While other works have claimed some relevance to flapping wing flying robots, there are major differences in approach: [15] was concerned with reinforcement learning of a fluid system that was a model of flapping flight whereas for the flapping wing robot of [16], motion control equations are not given. Rather, the focus was on investigation of optical flow on flapping wing flying robots. The Smartbird of [17] quite resembles our robot but only design-oriented research was undertaken. Autonomy was not achieved. By full autonomy, we imply a flying robot that can control its movements by itself and without any human control (a manual, remote-controlled operation is not considered autonomous as is often erroneously reported by researchers). Thus, instead of design and development, we are more concerned with au-

*Email: aneesh@toyota-ti.ac.jp

Specification	Value
Power source	11.1V 900mAh Lithium Polymer
Gross weight	430g
Length	78 cm
Wingspan	60cm (1 wing)
Motor	Pulso X2208/22 brushless dc motor
Servos	Hitec HS-56HB Karbonite Micro Servo (Elevator) Hitec HS-45 High Speed/Torque Servo (Rudder)
Radio	DMSS 2.4GHz TX module
Payload	160g
ESC	GWS ESC 15A
Sensor	APM 2.6 (3-axis gyro/acc., magnetometer, barometer), PIX4FMU

Table 1: Specifications of SlowHawk2 with articulated wings.

tonomous control. Therefore, this work is devoted to modelling and obtaining equations that govern the robot motion such that fully autonomous control can be implemented in the future. Our approach is therefore to use an existing and commercial flapping-wing platform and develop motion control algorithms for it. This subsequently entails a consideration of topics ranging from dynamical modelling and simulation to kinematics formulation and control.

This paper presents the first of our results. It presents the analysis and dynamical modelling of a bird-like flapping wing flying robot. The significant contribution of this paper is (i) the formulation of the dynamical model for a full-scale, outdoor-use flapping wing MAV with articulated wings and (ii) verification of dynamical model using both theory and *real experiments and real sensor data*. We first simulate the flying robot model in MATLAB to realize basic yawing and pitching motions which form the basis for future flight trajectories in 3D spatial space, and then repeat experiments with the real robot. Our intent is not to reverse engineer the flapping wing flying robot but to analyse it and derive a complete generic and scalable kinematics model for the overall genre of articulated-wing ornithopters. Compared to other works, our work focus on using a life-sized flapping wing flying robot with articulated wings intended for outdoor environments.

2 THE SLOWHAWK2 FLAPPING WING FLYING ROBOT

2.1 Design Analysis

The flapping-wing MAV under study is the SlowHawk2 flying robot¹ with articulated flapping-wings, shown in Figure 1. The original SlowHawk is available commercially and has been custom modified so that it has articulated wings, thereby being more life-like with increased payload. The full specifications are provided in Table 1.

A single brushless dc motor is the principal actuator responsible for the flapping mechanism. The motor is first connected to a gear train system of three gears $\{G_1, G_2, G_3\}$ for speed reduction. The last train gear G_3 drives a crank arm mechanism. The crank arm is coupled to a connecting rod which converts the motor circular motion into uniform flapping action (up and down flaps) of the two wings. This slider-crank configuration is labelled in the diagram of Fig-

¹SlowHawk2 with Articulated wings remodelled by Kazuhiko Kakuta

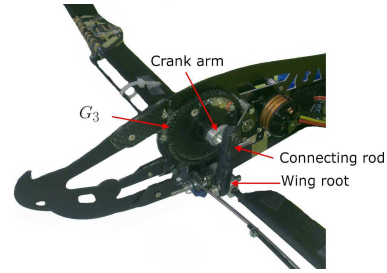


Figure 2: Slider-crank mechanism used to realize wing-flapping motion

ure 2. The brushless motor is used for actuating both wings simultaneously. With such concurrent flapping, one wing can not flap independently of the other. This flapping action contributes to the lift and propulsion of the robot.

The robot frame is made of carbon fiber frames and rods. The wing surface is made of thin, water-resistant vinyl tarpaulin. Two servo motors are used for actuation of the tail as a rudder and elevator. The rudder and elevator deflections control the steering (yaw) and pitching movements of the flying robot. Use of servos implies that the rudder and elevator deflections can be controlled by commanding them to move precise angles using internal feedback when controlled in position mode. Likewise, the instantaneous rudder or elevator position can be determined by reading the current servo position. This feature will be exploited later when controlling the robot autonomously.

A notable feature of the robot is the absence of any control mechanisms for rolling the flying robot. This implies that the robot makes a turn by controlling the tail rudder exclusively, thus without any active banking or rolling. This characteristic will later influence the number of derived equations for the dynamical model.

The payload of weight approximately 158g allows us to put various on-board electronics and sensors. Individual modules each weigh less than 50g so one or two units can be installed at a time for different measurements. Options for on-board sensors are included in the table of specifications.

2.2 System Modeling

Consider the brushless dc motor responsible for the flapping. Input to this motor is angular velocity in radians per second. Let us denote this as ω_b rad/s. Let η be the total gear ratio of the entire gear train $\{G_1, G_2, G_3\}$ system (found by taking the combined ratios of all individual gears, which is not shown here) The reduced gear speed, of the last gear G_3 in the gear train, is therefore

$$\omega_3 = \eta \omega_b \quad (1)$$

According to the well-known physics of a crank mechanism, the connecting rod coupled to the last gear moves up and down uniformly with a velocity given by

$$v = -r \omega_3 \left(\sin \theta + \frac{r \sin 2\theta}{2l \cos \beta} \right) = v_{flap} \quad (2)$$

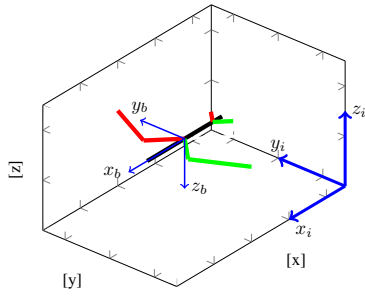


Figure 3: Assignment of reference frames

where θ is the crank angle from the vertical and

$$\cos \beta = \sqrt{(1 - (r/l)^2 \sin^2 \theta)}. \quad (3)$$

where r is the crank length and l is the connecting rod length. Since the connecting rod is coupled to the wing root and actuates it directly, this is the velocity with which the wings flap, i.e the speed of one flap v_{flap} , in terms of the input motor velocity. The frequency of flapping f_{flap} is therefore

$$f_{flap} = \omega_3 / 2\pi \quad (4)$$

3 DYNAMICAL MODEL

3.1 Assignment of Axes and coordinate Frames

Two axes frames are assigned to the system as shown in Figure 3. The **body** frame $F_b = \{x_b, y_b, z_b\}$ is assigned relative to the instantaneous position of the flapping wing flying robot by fixing the frame to its body. The frame origin is at the robot c.o.m with the z axis pointing vertically down. The **inertial** reference frame $F_i = \{x_i, y_i, z_i\}$ acts like a global coordinate system and is set with $z = 0$ on the ground plane and pointing upwards while the directions of the x and y axes coincide with that of the body frame.

3.2 Position and Orientation Representation

The pose of the flapping-wing flying robot is specified by a six d.o.f specification. It is given by $(x, y, z, \phi, \theta, \psi)$ where the first triplet (x, y, z) specifies the robot position in the global coordinate system and the second triplet (ϕ, θ, ψ) represents the orientation which is normally done using the x-convention for Euler angles (corresponding to roll, pitch and yaw). If A, B, C are the matrices that specify rotations corresponding to Euler angles (ϕ, θ, ψ) , then the orientation R is expressed as

$$R(\phi, \theta, \psi) = CBA \quad (5)$$

To circumvent the problem of the dreaded gimbal lock, the orientation is specified using quaternions instead of Euler angles in this work. Thus, the robot orientation is described as a rotation about an arbitrarily fixed axis using quaternions of the form

$$q = w + xi + yj + zk \quad (6)$$

with $i^2 = j^2 = k^2 = -1$. Here w describes the size of the rotation in $\mathbb{R}^3$ about the arbitrarily fixed and normalized axis defined by the vector x, y, z . Using this system, if the robot is oriented at an angle of θ about the axis defined by x, y, z , then the resulting quaternion is

$$q = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} xi + \sin \frac{\theta}{2} yj + \sin \frac{\theta}{2} zk \quad (7)$$

Once a quaternion describing a rotation or orientation in $\mathbb{R}^3$ is obtained using equation (7), the resulting gimbal-lock free, quaternion-derived rotation matrix representing the robot orientation is given by

$$qR = \begin{pmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2zw & 2xz + 2yw \\ 2xy + 2zw & 1 - 2x^2 - 2z^2 & 2yz - 2xw \\ 2xz - 2yw & 2yz + 2xw & 1 - 2x^2 - 2y^2 \end{pmatrix} \quad (8)$$

Then

$$R = qR \quad (9)$$

Using quaternions to represent orientation makes it possible to realize smoother rotations of the flapping wing flying robot. Although regular Euler-derived rotation matrices give the same end result for a rotation, the movement occurs in 'jerks'. This occurs because the final rotation matrix is a sequence of three successive rigid rotations: $R = \phi^\circ \rightarrow \theta^\circ \rightarrow \psi^\circ$. Quaternion-derived rotation matrices, on the hand, allow determination and execution of smoother rotations achieved using spherical linear interpolation. Euler angles are therefore only used at the output stage.

Spherical linear interpolation (*slerp*) [18] finds the quaternion matrix representing the smoothest rotation from orientation q_1 to q_0 using the surface of the unit hypersphere for the interpolation. This technique can be used to find the smoothest rotational manoeuvre to change to a certain orientation. For example, external disturbances during flight cause the robot to be disoriented to an orientation q_1 . We wish to again smoothly manoeuvre it back into its original orientation q_0 . Then we could compute the smoothest manoeuvres using *slerp* according to eq. (10) for $0 \leq t \leq 1$ seconds:

$$\cos \Omega = q_0 \bullet q_1$$

$$slerp(q_0, q_1, t) = \frac{q_0 \sin((1-t)\Omega) + q_1 \sin(t\Omega)}{\sin \Omega} \quad (10)$$

The resulting quaternion is the smoothest, constant-velocity manoeuvre from q_1 to q_0 that occurs in 1 second. To control the manoeuvre speed, the quaternion is simply time-scaled by multiplying with the desired time duration in seconds.

3.3 Dynamical Model

Treating the robot body as rigid, Newtonian mechanics provides the basis of developing motion equations that describe the motion and orientation of the c.o.g of the flapping wing flying robot as it flies. Net forces and net moments acting about the three axes are used to deduce the linear and

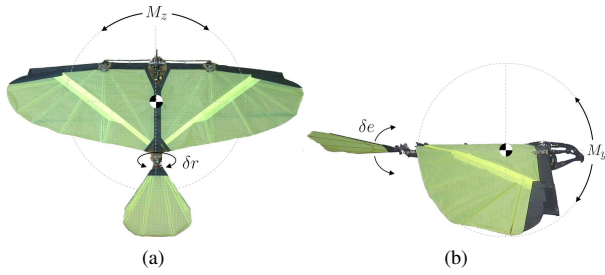


Figure 4: Moments induced by rudder (a) and elevator deflection (b).

angular accelerations which then enable determination of the linear and angular velocities and displacements, in their corresponding frames respectively.

We first consider the rotational component as that yields the robot orientation which is required for the linear modelling. Thus, the moments acting on the flapping wing flying robot are deduced. Short-term moments are induced on the robot due to the movement of each of the control surfaces like tail elevator and rudder. We suppose that when the tail elevator or the rudder moves, the robot pitches or yaws about its center of gravity, respectively as shown in Figure 4. Ignoring damping, short-term moments contributed by the movements of the tail elevator and rudder are

$$\begin{aligned} M_y &= 0.5 \rho V^2 S_{wing} \bar{c} (C_{y\delta e} \delta e) \\ M_z &= 0.5 \rho V^2 S_{wing} b (C_{z\delta r} \delta r) \end{aligned} \quad (11)$$

Intuitively, the equations of (11) quantifies the moments M_y and M_z induced in the y and z axes of the F_b frame by the elevator and rudder deflections δe , δr respectively. The dimensionless coefficients quantify or represent the individual net contribution of the deflecting control surface to the corresponding moment. That is, $C_{y\delta e}$ is the moment contribution of the elevator deflection δe (in degrees) to the moment in the y axis and $C_{z\delta r}$ is the moment contribution of the rudder deflection δr (degrees) to that of in the z axis. $\bar{c}$ is the wing mean chord and b is the wing span. The equations indicate *what* control surface contributes to *which* moment. As one would expect, the elevator deflection δe causes the pitching moment M_y whereas the rudder deflection contributes to the yawing moment M_z . Recall that the flying robot is incapable of banking actively in flight due to the absence of any relevant control surface. This explains the exclusion of any active rolling moments M_x in the equations.

Gravity contributes zero moments. Further, the influence of the flapping wings to moments induced depends on the gyro sensor placement. If a gyro sensor is placed on a wing, its moving crank arm mechanism or any other location that moves as a result of the wings flapping, naturally it will register rotational motion. However, if the gyro sensor is placed over the c.o.g, no rotational motion is registered since they only effect of the flapping wings upon the c.o.g is propelling it forward and lifting it up. Thus, only moments M_y and M_z are responsible for overall angular accelerations (and

orientation) of the robot body in the F_b axes. The rotational acceleration equations are therefore

$$\begin{aligned} I_{yy} \ddot{\theta} &= M_y \\ I_{zz} \ddot{\psi} &= M_z \end{aligned} \quad (12)$$

where I is the moment of inertia of the robot. The angular accelerations $\ddot{\phi}$, $\ddot{\theta}$ and $\ddot{\psi}$ in the body frame are then easily determined. The angular velocities are

$$\dot{\phi}(t) = 0 \quad \dot{\theta}(t) = \int_0^t \ddot{\theta}(t) dt \quad \dot{\psi}(t) = \int_0^t \ddot{\psi}(t) dt \quad (13)$$

Conditions must be imposed in the integration process above to resolve ambiguities due to the dual connotations of acceleration:

1. Zero acceleration implies zero velocity (and not constant velocity)
2. Decreasing acceleration implies magnitude of velocity decreases to zero (and not to a constant value)
3. Negative acceleration implies decreasing velocity in same direction (and not an acceleration in the opposite direction)

Finally the quaternion representing the orientation can be determined from the angular velocities. Given the angular velocity is related to the derivative of the quaternion as

$$\omega = \begin{bmatrix} 0 \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = 2 \frac{dq}{dt} \otimes \tilde{q} \quad (14)$$

the quaternion representing the orientation can therefore be determined

$$q = \int_0^t \frac{1}{2} \omega dt \otimes \tilde{q} \quad (15)$$

Now consider the free body diagram (f.b.d) of the flapping wing flying robot in steady-state flight in Figure 5. By “steady-state”, flight conditions of forward direction at constant velocity, constant altitude and zero pitch and roll are implied. The major forces acting on the flapping wing flying robot are shown: lift L , weight mg , thrust T , drag D and disturbances $F_{disturb}$. According to [19], a whole aircraft can be treated as a single entity when modeling its forces. Therefore, the equal but separate thrusts and lifts generated on either side of the flapping wing flying robot by its flapping wings are treated as one individual lift and thrust vector acting along and on the c.o.m without any significant loss of precision. The combined wing lift is accordingly shown as $L = L_1 + L_2$ acting over the robot c.o.m. Likewise, the total thrust generated by the two wings $T = T_1 + T_2$ is similarly shown acting forward along the x -axis.

Various research groups in aerodynamics have studied the physics and dynamics of flapping-wing flight and computed the thrust and lift generated [20, 21, 22]. However, most of the computations involve wing-specific parameters with specific experimental conditions, assumptions and test-beds. Accordingly, they cannot be unconditionally applied in this work. For example, we refer to the work of [22]. In the analysis of lift generation of a hummingbird wing model, the direction of the lift vector was taken to always point vertically upwards irrespective of the robot orientation. This is attributed to the active torsioning of the wings when flapping whereby the direction of the lift vector varied with the torsioning. However, the wings of the SlowHawk2 do not undergo any torsioning. The lift vector is therefore always remaining perpendicular to the robot body. Furthermore, instead of deriving design models as done in [6] and [7], our objective is to formulate those equations that describe the *motion* of the robot c.o.g as it flies. Consequently, instead of design-oriented direct calculation of thrust and lift, the resultant lift and thrust of the flying robot is deduced. Without any loss of precision, the net thrust T is given by

$$T = \frac{P}{V} \quad (16)$$

by calculating the net power P of the robot using the specifications of the brushless motor and measuring the velocity V of the flying robot in steady flight beforehand. In [22], the same formulation was used to estimate the lift although we give a different approach for lift estimation. It is recognized that the Lift force produced due to the flapping actions will be function of the speed of the flapping wings. Recall that the velocity of a wing flap has been computed. Thus for lift estimation,

$$L = 0.5 \rho v_{flap}^2 S_{wing} C_l \quad (17)$$

Equation (17) is derived from fluid dynamics theory used for calculating forces, in this case lift, acting upon aerial vehicles. Here ρ is the air density, S_{wing} wing surface area and C_l the lift coefficient. v_{flap} is the wing flap velocity calculated in (2) and as long as v_{flap} can be calculated for other types of flapping-wing flying robots, the above expression for lift may be used. The flying robot is to be controlled at constant altitude and velocity so all parameters in this equation remain unvarying. Drag is given by

$$D = 0.5 \rho V^2 S C_d \quad (18)$$

where V is flying robot velocity, S body surface area and C_d the drag coefficient.

Given that a flapping wing flying robot is lighter than other types of MAV's, its motion is consequently more susceptible to disturbances such as wind gusts and Coriolis effect etc. Therefore, all probable disturbance forces are collectively modelled as $F_{disturb}$. Although MATLAB provides some built-in wind models that can be used to model disturbances, these are all designed to military-level specifications.

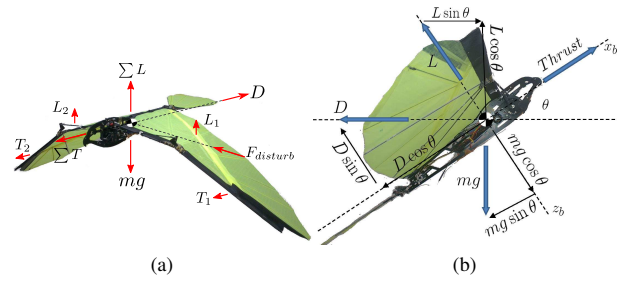


Figure 5: Forces acting on flapping wing flying robot when in flight.

Thus, we define a disturbance force ourselves, in the inertial frame, as a vector-valued, quaternion-based force vector as $F_{disturb} = \begin{pmatrix} k \\ q \end{pmatrix}$ where k is the disturbance force magnitude and q the quaternion representing direction relative to the inertial frame. If the robot has some orientation q_0 relative to the inertial frame, the effective direction of $F_{disturb}$ acting on the robot is determined through straightforward quaternion multiplication.

$$F_{disturb}\{q\}|_{\text{effective}} = q_0 \times q_{disturb} \quad (19)$$

Thus, for a flapping wing robot flying oriented with some roll ϕ , pitch θ , and say for example, $F_{disturb}$ effectively acting along the y axis as illustrated in Figure 5, the equations for linear accelerations in each of the axes of the F_i frame are given by

$$\begin{aligned} m\ddot{x} &= \cos \theta \cos \psi \sum_{i=1}^2 T_i - D \cos \theta \cos \psi - \sin \theta \sum_{i=1}^2 L_i \\ m\ddot{y} &= F_{disturb} + \sin \psi \sum_{i=1}^2 T_i - D \sin \psi \\ m\ddot{z} &= \sin \theta \sum_{i=1}^2 T_i - D \sin \theta + \cos \theta \sum_{i=1}^2 L_i - mg \end{aligned} \quad (20)$$

The kinematic equations are

$$\begin{aligned} \dot{x} &= V \cos \theta \cos \psi \\ \dot{y} &= V \cos \theta \sin \psi \\ \dot{z} &= V \sin \theta \end{aligned} \quad (21)$$

The motion and orientation of the flapping wing flying robot is therefore described by equations (20), (12), (15) and (21). It is important to state that although acceleration data obtained from on-board sensors will be expected to have white noise and bias, two solutions circumvent this dilemma. First is the use of a high pass filter to remove low frequency noise in the acceleration data which is responsible for integration drift errors. Alternatively, the acceleration data can be neglected altogether since it is the rotational velocity that is used in determining the relative orientation and this can directly be acquired from the gyro sensor. Further, rotational velocity information is primarily intended for executing momentary

manoeuvres in changing flight direction and path only. These momentary corrective manoeuvres occur in the order of milliseconds or an absolute maximum of 1-2 seconds. Therefore, accumulative errors of the gyro will be at a minimum. For position measurement, it is envisaged that absolute positioning systems such as GPS for outdoor use and motion capture systems for indoor use will be employed.

4 EXPERIMENTAL RESULTS

Quaternion SLERP Test. A demonstration of using spherical linear interpolation to determine a smooth correctional manoeuvre is given in the accompanying video² (result is given as a video due to the *motion* involved). The robot model is initially shown in its original, steady-state configuration with orientation q_0 and the disturbed orientation, due to external disturbance forces, as q_1 . The video contains the smooth correctional manoeuvre from q_1 to q_0 as determined using quaternion-based spherical linear interpolation from eq. (10).

Model Validation. Equations for the dynamical model were validated by supplying separate input signals for the elevator and rudder deflections that would produce pitching and yawing motions, both in simulation and in real experiments. Motion data generated during simulation was then compared to real sensor data. As a proof-of-concept simulation, constant coefficients in the equations were defined as symbolic expressions since their values are yet unknown. A consequence of this approach is the axes of certain graphs are not quantified discretely, instead they exhibit the general motion profile which suffices to represent the robot behaviour emanating from the control input. For a robot flying in steady-state condition at arbitrary altitude and speed, the commanded input signal applied to the elevator and rudder deflections to realize pitching and yawing was the trapezoidal curve with amplitude A and zero length upper-base specified with the function

$$f(t; A, a, b, c) = \begin{cases} 0 & \text{if } t \leq a \\ A \frac{t-a}{b-a} & \text{if } a \leq t \leq b \\ A & \text{if } t = b \\ A \frac{c-t}{c-b} & \text{if } b \leq t \leq c \\ 0 & \text{if } t \geq c \end{cases}$$

by the definition of which the control surface deflection commences at $t = a$ steadily, reaches maximum deflection of A° at $t = b$ after which it retracts and returns to zero deflection position by $t = c$. The rate of deflection is therefore A°/s . A visualization of this is given in Figure 6.

Pitching Test. The signal specified for the elevator deflection in this test was $f(t; 10^\circ, 1, 2, 3)$ and for the rudder $f(t; 0^\circ, 1, 2, 3)$ i.e zero deflection.

The complete motion data is shown in Figure 7 for the robot body in the body frame of reference F_b . The angular acceleration data was obtained by applying eq. (11) and

eq. (12) using the commanded elevator deflection function $f(x; A, 1, 2, 3)$ as input and adding simulated white Gaussian noise. It is seen that computed motion data subsequently derived from acceleration data correctly characterize the robot pitching motion at every instant of the robot movement apart from minor anomalies that are consequences of the simulated white Gaussian noise. For the first 1 second, the robot is in steady state flight. At the $t = 1$, robot starts pitching while continuing to fly forward. At $t = 2$, the robot halts pitching downwards and starts reverting back to zero pitch position as a consequence of the elevator deflection being gradually reduced. From $t = 3$ onwards, the robot has resumed steady-state condition albeit having lost altitude due to the pitching motion.

Inspecting the visualizations of the simulated rotational motion data in Figure 7, as the robot pitches forwards(downwards) around the y axis, the angular acceleration increases linearly (with minus sign). The corresponding angular velocity increases parabolically while displacement increases exponentially. Velocities around the x and z axis remain unchanged as there is no active rolling or yawing. The velocity curves appears to be free of white noise jitter but this is a superficial effect of integrating acceleration data using the trapezoidal method that has a smoothing effect on the resulting curves.

For the simulated linear motion:

1. z acceleration of the body frame increases linearly in the time interval the robot is pitching forward due to direction of the thrust vector gradually turning towards direction of gravitational pull that exists in the z axis. Same is also true for the y velocity.
2. The linear x acceleration decreases from zero acceleration while the robot was in steady state to a negative value in the time the robot pitches since the acceleration in that direction is reduced momentarily while the robot pitches and rescinds pitching. Consequently, there is also a drop in the x velocity in this interval.
3. y acceleration is unvaried.

The resultant linear velocities and displacements then depict the linear motion of the robot. The change in altitude as a consequence of the pitching motion is clearly seen in the z axis displacement curve in the displacement graph. Finally, the same experiment was repeated by flying the real robot non-autonomously to log sensor data. Comparing the gyro sensor data acquired in Figure 9(a), the Gyro Y data exhibits the same parabolic form as the simulated velocity data in Figure 7 when the robot underwent pitching.

Yawing Test. The specified rudder input command was $f(x; A, 4, 5, 6)$. The simulated results are shown in Figure 8 and the real sensor data is given in Figure 9(b). The consequence of this rudder deflection is the robot yawing to the left and then realigning to be parallel to the original flight direction (see “Y” curve in linear displacement graph). At the mo-

²<http://aneeshchand.webatu.com/imav2014.html>

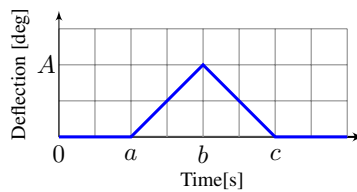


Figure 6: Input signal format for control surface deflection

ment the robot yaws, the real gyro sensor data again exactly replicates the simulated data.

Behaviour of the robot from both theory and experiments indicated a non-linear relationship between control surfaces' deflection and corresponding motion. For instance, for small values of δr or δe , the robot yaw or pitch magnitude is small but as the deflection is increased, the motion changes non-linearly. This was evident from the results: as the angular acceleration increased linearly, the angular rate increased parabolically while the angular displacement was exponential. This indicates that any future controllers designed would strictly be non-linear. Videos in the provided link show one of the simulations and a short demonstration of the real robot.

5 CONCLUSION

A simplified model describing the motion of a flapping wing flying robot with articulated wings was formalised with flapping averaged dynamics. Quaternions were used to represent the robot orientation and acquire smooth rotations in 3D space. Force-moment equations were then used to derive the dynamical equations that governs the robot motion. Simulations revealed the robot to respond to external stimuli in the form of elevator and pitch input deflections with simulated data qualitatively corroborating with real sensor data. Some constraints were assumed: only the salient active forces and moments were considered in the dynamics. Induced or residual forces and moments from body vibrations and flapping were ignored in the first prototype. Since only short-range navigation in the order of tens or hundreds of meters is considered, Earth's rotation and Coriolis acceleration were neglected. Likewise, only low altitude flying is considered so air-density was assumed to be non-varying. Given that the dynamical and kinematic model for the flapping-wing flying robot has been realized, the immediate next step is to design methods of estimating values of the parameters and coefficients. For reference position measurement, motion capture systems and/or GPS will be utilized.

Finally, given that some of the dynamics will be highly time varying, our future intention is to use adaptive control techniques for autonomous control where the on-line parameter estimation component of the adaptive controller would counter any deficiencies in the model.

ACKNOWLEDGEMENTS

The authors thank Dr. Kazuhiko Kakuta of Miyagi, Japan, developer of SlowHawk2 for kindly supplying the robot.

REFERENCES

- [1] Kenzo Nonami, Farid Kendoul, Satoshi Suzuki, Wei Wang and Daisuke Nakazawa, "Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles", Springer, 2010.
- [2] European Commission, "Towards a European strategy for the development of civil applications of Remotely Piloted Aircraft Systems (RPAS)", Brussels, September 2012.
- [3] Lorenz Meier, Petri Tanskanen, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer and Marc Pollefeys, "PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision", Autonomous Robots (2012) 33:2139.
- [4] M. Hehn and R. D'Andrea, "Quadcopter trajectory generation and control", In IFAC World Congress, pages 1485-1491, 2011.
- [5] 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: 061006, 2009.
- [6] Madangopal, R., Khan, Z. A., Agrawal, S. K. "Biologically inspired design of small flapping wing air vehicles using four-bar mechanisms and quasi-steady aerodynamics", Journal of Mechanical Design, Vol. 127, No.4, 2005, 809-816.
- [7] Grauer, J., Hubbard, J.E., "A Multibody Model of an Ornithopter, AIAA Paper No. 2009-727.
- [8] de Croon, G.C.H.E., M.A. Groen, De Wagter, C., Remes, B.D.W., Ruijsink, R., and van Oudheusden, B.W. "Design, Aerodynamics, and Autonomy of the DelFly" In Bioinspiration and Biomimetics, 2011.
- [9] Percin, M., Hu, Y., van Oudheusden, B.W., Remes, B., and Scarano, F. "Wing flexibility effects in clap-and-fling" International Journal of Micro Air Vehicles, Volume 3, Number 4, pp. 217-227, 2011.
- [10] Tay, W., Hester, B., van Oudheusden, B. "Analysis of biplane flapping flight with tail", at the 30th AIAA Applied Aerodynamics Conference, AIAA-2012-2968, 2012.
- [11] de Croon, G.C.H.E., de Clercq, K.M.E., Ruijsink, R., Remes, B., and de Wagter, C. "Design, aerodynamics, and vision-based control of the DelFly" In the International Journal of Micro Air Vehicles, Volume 1, Number 2, pp. 71 - 97.
- [12] Robert J. Wood, "The First Takeoff of a Biologically Inspired At-Scale Robotic Insect", IEEE Transactions on Robotics, Vol. 24, No. 2, April, 2008, pp. 341-347.
- [13] Benjamin M. Finio and Robert J. Wood, "Open-loop roll, pitch and yaw torques for a robotic bee", IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct. 7-12, 2012, Portugal.
- [14] Xinyan Deng, Luca Schenato, Wei Chung Wu and Shankar Sastry, "Flapping Flight for Biomimetic Robotic Insects Part I: System Modeling", IEEE Transactions on Robotics, Vol. 22, No. 4, April, 2006, pp. 776 - 788.
- [15] John W. Roberts, Jun Zhang, and Russ Tedrake, "Motor learning at intermediate reynolds number: Experiments with policy gradient on the flapping flight of a rigid wing", In From Motor to Interaction Learning in Robots. Springer, 2009.
- [16] Fernando Garcia Bermudez and Ronald Fearing, "Optical flow on a flapping wing robot", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5027 - 5032, St. Louis, 10-15 Oct. 2009.
- [17] W. Send, Markus Fischer, Kristof Jebens, Rainer Mugrauer, Agalya Nagarathinam and Felix Scharstein. "Artificial Hinged-Wing bird with Active Torsion and Partially Linear Kinematics", 28th International Congress of the Aeronautical Sciences, Brisbane 2012.
- [18] Edward Pervin and Jon A. Webb. "Quaternions in Computer Vision and Robotics". Carnegie-Mellon University, 1992.
- [19] David Allerton, "Principles of Flight Simulation", Wiley, pp. 47, 2009.
- [20] W. Send, "Flapping-Wing Thrust in Compressible Flow" 25th International Congress of Aeronautical Sciences (ICAS), Hamburg (Germany), September 3-8, 2006.
- [21] D. Mueller, H.A. Bruck, and S.K. Gupta. "Measurement of thrust and lift forces associated with drag of compliant flapping wing for micro air vehicles using a new test stand design". Experimental Mechanics, 50(6):725735, 2010.
- [22] Hiroto Tanaka, Haruka Suzuki, Ikou Kitamura, Masateru Maeda and Hao Liu, "Lift Generation of Hummingbird Wing Models with Flexible Loosened Membranes", IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Nov. 3-8, 2013.

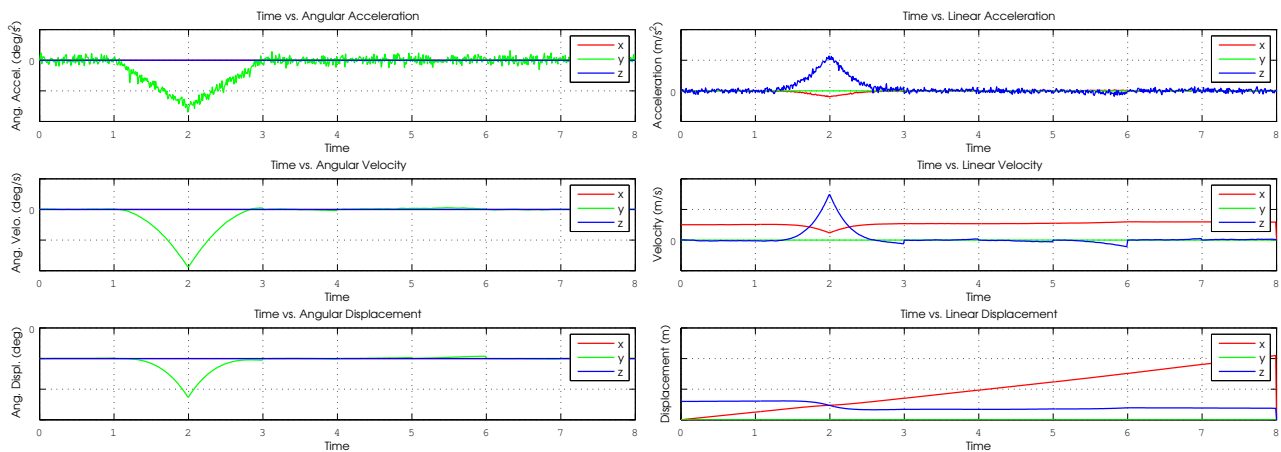


Figure 7: Simulated rotational and linear motion data for pitching movement. The “Z” curve of the linear displacement plot reveals the altitude change as a consequence of the downwards pitching motion

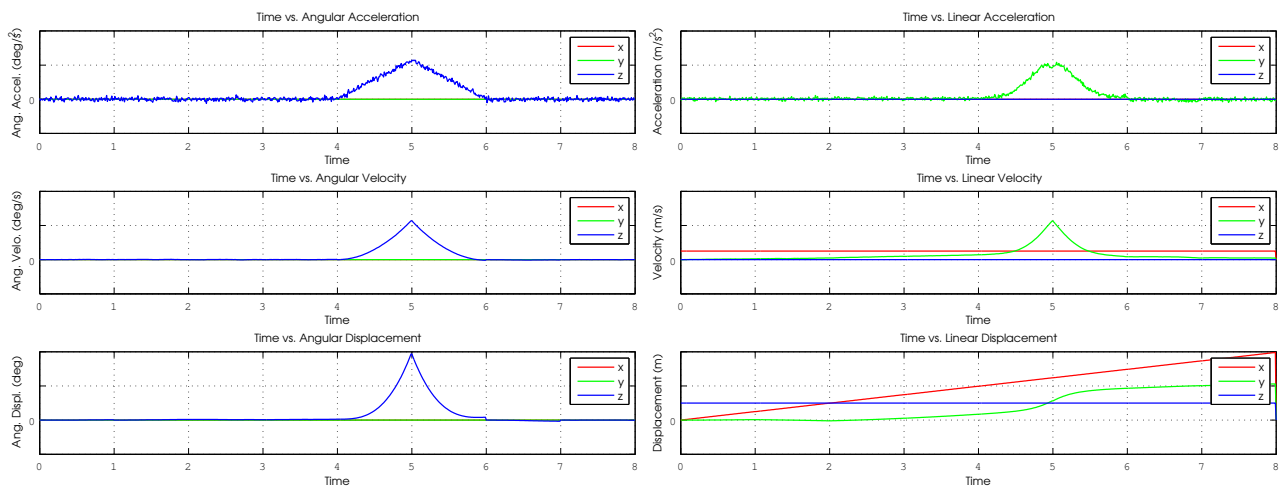
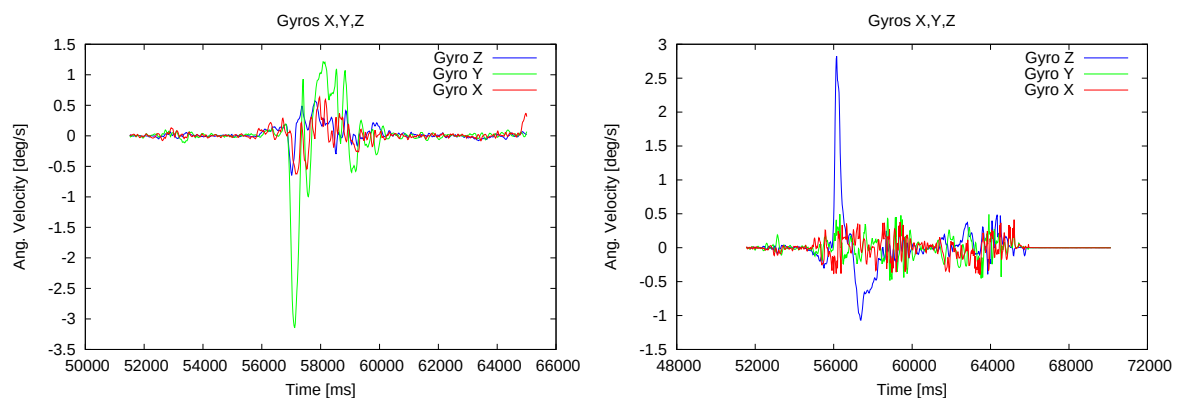


Figure 8: Simulated rotational and linear motion data for yawing movement. The change in flight direction is discernible from the “Y” curve of the linear displacement plot



(a) Real gyro sensor data for pitching motion. Aside from spurious data, it is seen that the Gyro Y data exhibits an exponential curve at approx. $t = 57000$ when the real robot pitched. This matches with simulated data of the angular velocity graph in Figure 7 at $t = 2$

(b) Real gyro sensor data for yawing motion. Again ignoring spurious data that can be easily filtered, note the overall resemblance of the Gyro Z exponential curve at $t = 56000$ when the real robot yawed to simulated data in the angular velocity graph of Figure 8 at $t = 5$

Figure 9: Real gyro sensor data for pitching (a) and yawing (b) motions

Off-board Visual Odometry and Control of an Ultralight Quadrotor MAV

Kun Li*, Rui Huang, Swee King Phang, Shupeng Lai, Fei Wang, Ping Tan, Ben M. Chen and Tong Heng Lee
Department of Electrical and Computer Engineering, NUS, Singapore, 117576

ABSTRACT

In this paper, we propose an approach to autonomously control a quadrotor micro aerial vehicle (MAV). With take-off weight of 50 g and 8-min flight endurance, the MAV platform codenamed 'KayLion' developed by the National University of Singapore (NUS) is able to perform autonomous flight with pre-planned path tracking. The vision-based autonomous control is realized with a light weight camera system and an ultrasonic range finder integrated to the MAV. An optical flow algorithm is adopted and processed on ground control station to provide position and velocity estimation of the MAV. A model-based position controller is implemented to realize autonomous flight.

1 INTRODUCTION

The rapid development in the area of unmanned aerial vehicles (UAVs) has seen breakthrough and advancement of small-scaled aerial vehicles. These small-scaled air vehicles could be used as scouts in many dangerous civil and military missions, especially showing their superiority in cluttered and constrained indoor environments [1, 2, 3]. Moreover, along with the progress of microchip and microelectromechanical systems (MEMS) technologies, the size of the UAVs have been scaled down to centimeter level. As an example, a palm-sized gliding MAV developed by Harvard University [4], weighing 2 g and 10 cm in length, is capable of autonomous flight target sensing and obstacle avoidance with an optical flow sensor. Another example is Robobee [5], a flapping wing platform created by the same research group. It is only 83 mg and is capable of lift off with external power and execute open-loop pitching and rolling maneuvers. Despite the fact that the platform is only able to fly within a certain distance from the power supply, it is a breakthrough in miniature aerial vehicles.

The term "optical flow" [6] is a bio-inspired concept. It relates to the pattern of apparent motion of objects, surfaces and edges in a visual scene caused by the relative motion between an observer, which is a mono-camera in this case, and the scene. The optical flow method is essential to navigation strategy of MAV due to the poor performance of the

low-cost MEMS GPS sensor signal, or even in indoor environments without GPS. In [7], an optical flow based technique is adopted in outdoor navigation to aid GPS and INS measurement. [8, 9, 10] demonstrated feasible approaches to autonomous MAV navigation and localization in GPS-denied environments with optical flow based motion estimation. The MAV is able to perform autonomous flight in unknown environment with all the algorithms running onboard, while this is not applicable to the ultra-light platforms as described in this paper. In the work done by TU Delft [11], Delfly, a flapping-wing platform with 16 g gross weight including sensors, microcontroller and analog camera, achieved autonomous flight indoor. These works serve as excellent examples to illustrate feasible approaches for indoor and GPS-denied environment navigation, especially implemented for MAV.

This work is an extension of our previous work [12] with vision-based position control and path following. Section 2 discusses the hardware selection, design and assembly with detail specifications. In Section 3, an optical flow based algorithm for motion estimation is presented. Section 4 gives the design methodology of inner-loop and outer-loop control along with the path generation, followed by flight test results provided in Section 5. Concluding remarks are made in Section 6.

2 PLATFORM DESIGN

In this section, a brief introduction of both the hardware design of quadrotor platform and the closed-loop autonomous control will be presented.

2.1 Quadrotor Platform

Our previous work [12] described an explicit methodology to design the quadrotor MAV platform, including mechanical analysis and design, electrical design and assembly as well as test-bench experiments for parameter identification. As shown in Fig. 1a, the quadrotor MAV codenamed 'KayLion', weighing 44 g and 15 cm of the diagonal length, consists of a bare quadrotor frame, an attitude and heading reference system (AHRS), as well as a 360 mAh Lithium-Polymer (Li-Po) battery. The system is orientation-stabilized by its onboard inertial measurement unit (IMU) feedback control, with an updating rate of 100 Hz.

In the block 'UAV' in Fig. 2, structure of the quadrotor MAV avionic system is displayed. The micro processor Atmega328p receives signals from the 2.4 GHz receiver, angle and angular measurements from the AHRS system and

*Email address: kunli89@nus.edu.sg

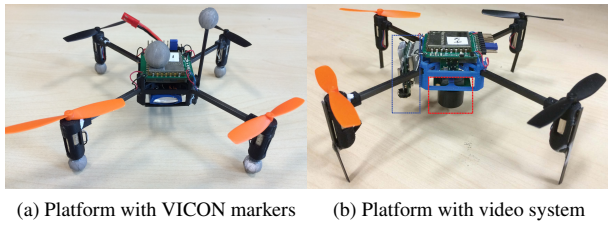


Figure 1: Quadrotor MAV platform codenamed 'KayLion'

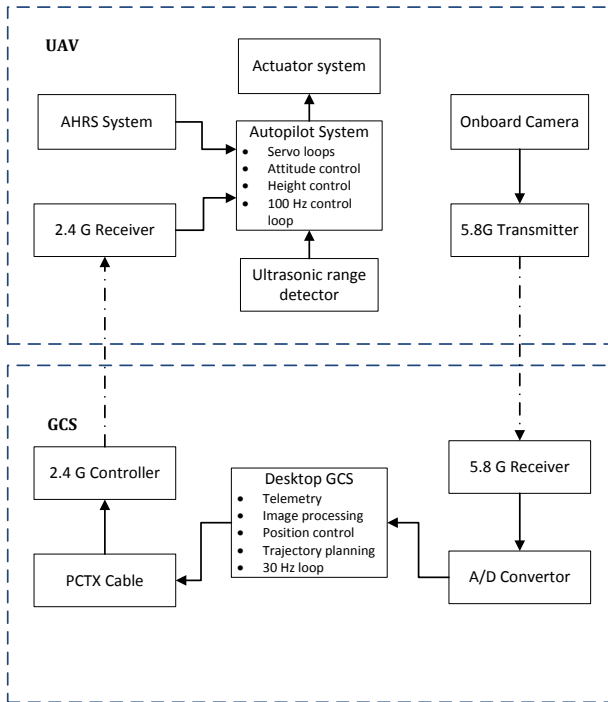


Figure 2: Autonomous control loop hardware architecture

output pulse-width modulation (PWM) signals to four electrical speed controllers (ESCs) to actuate the system.

2.2 Autonomous Control Architecture

Due to limited payload of the quadrotor MAV, there is no weight budget for high power processing units. Thus, the ground control station (GCS) is indispensable for the closed-loop control. Fig. 2 shows the overall architecture of the autonomous UAV, including the orientation stabilization and the position control.

2.2.1 Camera Design

A camera system consisting of an analog camera, an analog transmitter and a 40 mAh Li-Po battery to provide an 8 min flight is customized. Table 1 lists the specifications of the analog camera. The specifications of the analog video trans-

Table 1: Specifications of camera shown in Fig. 3a

Descriptions	Specs
Weight (g)	2
Resolution	704×576
Pixel amount	400000
Lens (mm)	2.78
Power supply (V)	3.5 - 5
View angle	62°

Table 2: Specs of video transceiver shown in Fig. 3b

Descriptions	Transmitter	Reciever
Module	TX5813	RX5808
Weight (g)	3.4	6.5
Frequency range (MHz)	5705-5945	5705-5945
Power supply (V)	3.5-5	3.5-5
Dimension (mm)	$22 \times 20 \times 3$	$28 \times 23 \times 3$

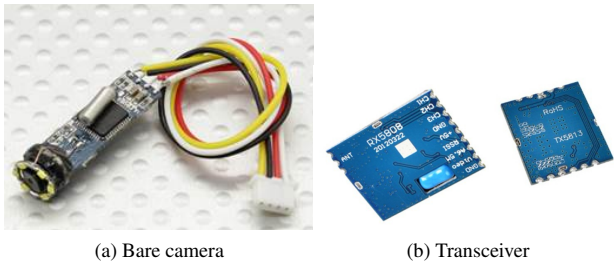
mitter and receiver are highlighted in Table 2. A 40 mAh capacity Li-Po battery was selected to provide sufficient power supply to the camera system. They are integrated on a single customized printed circuit board (PCB) for easy mounting on the platform. Experiment shows that the battery is able to power the system for more than 8 minutes. Fig. 3a shows the mini analog camera, Fig. 3b displays the transceiver combo, Fig. 3c shows the battery and Fig. 3d shows the integrated video system.

2.2.2 Ultrasonic Sonar Sensor

In order to maintain the aircraft at a certain height to perform stable indoor navigation, an ultrasonic sensor, Maxsonar EZ4, is adopted to provide altitude measurement. This sensor provides accurate readings from 0.15 m to 7.65 m with 1 cm resolution and can be powered with 3.3 V power supply. In Fig. 2, ultrasonic sensor is connected to the onboard avionic system via an analog-to-digital port. The overall assembly is presented in Fig. 1b, where the ultrasonic sensor locates in the center of the platform facing downward and the camera locates at rear side also facing downward.

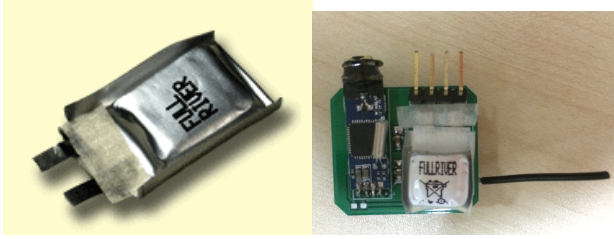
2.2.3 PCTx Connector

For the data-link solution, PCTx cable (see Fig. 4a), connecting the GCS to the RF controller, is a simple approach to meet the closed-loop control without adding extra wireless modules to the MAV. The cable provides 9 channel communication of 50 Hz pulse-position modulation (PPM) signal with 1024 steps resolution. The PCTx cable is able to transmit



(a) Bare camera

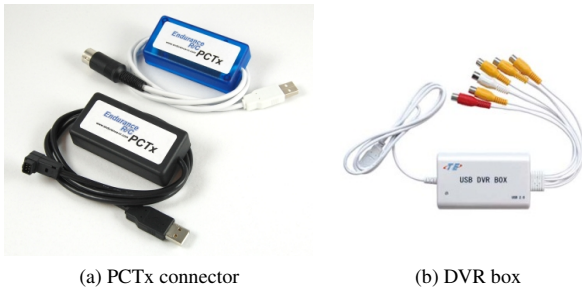
(b) Transceiver



(c) Camera battery

(d) Video system

Figure 3: Components for video system



(a) PCTx connector

(b) DVR box

Figure 4: Components for autonomous control

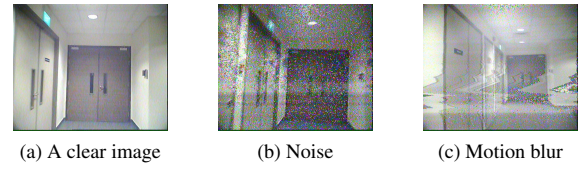
controlled signal generated by the GCS to the aircraft through the RF transmitter.

2.2.4 A/D converter

As indicated in Fig. 2, since the GCS cannot process analog video, a digital video recorder (DVR) (see Fig. 4b) is used to connect the receiver RX5808 to the GCS. The DVR box sends processed digital video signals to GCS via a USB port. A driver program for DVR box is installed on GCS to capture the digital video signal for image processing. The image resolution is set to be 704×576 pixels.

3 MOTION ESTIMATION USING OPTICAL FLOW

Optical flow is a well known algorithm in computer vision society to estimate the 2 dimensional (2-D) motion of features between two consecutive images. By examining the results of optical flow, we can further infer the 3 dimensional (3-D) motion of the camera. Optical-flow-based motion estimation is more memory efficient, compared to a full 3-D



(a) A clear image

(b) Noise

(c) Motion blur

Figure 5: Images affected by transmission noise and motion blur

reconstruction approach such as a visual SLAM system [13] which maintains a full 3-D map and feature tracks. Only the matched features between two consecutive frames need to be maintained in the proposed approach. To improve the computational speed, we downsize the image to 1/4 of the original resolution. Optical flow is further computed for a limited number of sparse feature points so that the system can run in real time at 25 Hz. Moreover, due to the severe transmission interference and motion blur in the video shown in Fig. 5, the feature tracking of the visual SLAM approach [13] might not work well. However, no feature tracks are needed in our case and false estimation due to images of low quality can be easily detected and then rejected. This makes optical-flow-based approach a suitable solution for motion estimation of the current platform.

In this section, we propose an approach to estimate the self motion of the quadrotor MAV equipped with a downward-looking camera via optical flow in an indoor environment.

3.1 Feature Detection and Homography Estimation

Firstly, feature points are detected using Shi-Tomasi feature detector [14]. The features are then tracked over consecutive frames using the Pyramid version of the Lukas-Kanade tracker [15]. In order to achieve high computational speed, we limit the maximum number of detected features to 100. Since the floor of the navigation area is poorly textured, we placed some random markers on the ground to provide sufficient features.

Given the corresponding feature points detected in consecutive frames, the perspective transformation between the two camera frames can be estimated. This transformation is represented by a 3×3 homography matrix. The locations of corresponding feature points $\mathbf{p}$ and $\mathbf{p}'$ in images captured at t and t' are related by the homography matrix as equation 1. $\mathbf{p}$ and $\mathbf{p}'$ are denoted as homogeneous coordinates.

$$s\mathbf{p} = \mathbf{H}\mathbf{p}' \quad (1)$$

According to [16], this relation is satisfied provided that the image scenes lie on the same plane. This assumption can be safely made in our case since all the captured scenes belong to the flat floor. The homography matrix $\mathbf{H}$ can be solved using a standard least square optimization algorithm. RANSAC [17] can be applied to reject outliers for a robust

estimation. The estimated homography is then refined using Levenberg-Marquardt optimization to further reduce the re-projection error. The homography matrix is estimated up to a scale s . Thus it needs to be normalized so that the element in homography matrix $\mathbf{H}_{33} = 1$.

3.2 Self Motion Estimation

As presented in [18], the homography matrix can be further decomposed as the following equation:

$$\mathbf{H} = \mathbf{R} + \frac{1}{d} \mathbf{T} \mathbf{N}^T \quad (2)$$

where $\mathbf{R}$ and $\mathbf{T}$ are the rotation and translation of the UAV frame from t to t' . $\mathbf{N}$ is the unit normal vector of the ground plane at t and d is the distance between the ground plane and the UAV frame. If we can obtain UAV attitude angles ϕ, θ, ψ at t and ϕ', θ', ψ' at t' , $\mathbf{N}$ can then be expressed as

$$\mathbf{N} = \begin{pmatrix} -\sin \theta \\ \sin \phi \cos \theta \\ \cos \phi \cos \theta \end{pmatrix} \quad (3)$$

The rotation $\mathbf{R}$ can be computed as follows,

$$\mathbf{R} = \mathbf{R}_{b/n}(t) \mathbf{R}_{n/b}(t') \quad (4)$$

where $\mathbf{R}_{b/n}(t)$ is the rotation from the UAV body frame to the inertia frame at t and $\mathbf{R}_{n/b}(t')$ is the rotation from the inertia frame to the UAV body frame at t' . Therefore, the translation $\mathbf{T}$ can be calculated as

$$\mathbf{T} = d(\mathbf{H} - \mathbf{R})\mathbf{N} \quad (5)$$

For general cases, we need the measurements from the IMU to calculate the $\mathbf{R}$ and $\mathbf{N}$. However, since the MAV is moving slowly during the navigation, it has negligible rotation. Thus we can safely assume that $\mathbf{R}$ is an identity matrix and $\mathbf{N}$ is the perpendicular vector to the ground plane. Also the distance between the ground plane and the MAV is measured by an ultrasonic sensor. As the translation $\mathbf{T}$ is estimated, the velocity vector $\mathbf{v}$ of MAV body frame can be computed by,

$$\mathbf{v} = \frac{\mathbf{T}}{\Delta t} \quad (6)$$

where $\Delta t = t' - t$.

3.3 Linear Kalman Filter

We further design a simple linear Kalman Filter to estimate the position of the MAV. The measurement is the estimated velocity based on the homography. The linear state space includes the position and velocity of the MAV,

$$\mathbf{x}_k = \begin{pmatrix} x \\ \dot{x} \end{pmatrix} \quad (7)$$

The state $\mathbf{x}$ is evolved according to the prediction model,

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \quad (8)$$

where

$$\mathbf{F}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \quad (9)$$

and

$$\mathbf{B}_k = \begin{pmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{pmatrix} \quad (10)$$

and $\mathbf{u}_k$ is the system input which is the MAV acceleration. We assume that the MAV undergoes a constant acceleration which follows a zero mean normal distribution. $\mathbf{w}_k$ is the process noise which follows $N(0, \mathbf{Q}_k)$.

The measurement model is,

$$\mathbf{z}_k = \mathbf{G}_k \mathbf{x}_k + \mathbf{v}_k \quad (11)$$

where

$$\mathbf{G}_k = \begin{pmatrix} 0 & 1 \end{pmatrix} \quad (12)$$

and $\mathbf{v}_k$ is measurement noise which follows $N(0, \mathbf{R}_k)$.

The optical flow algorithm will sometimes fail if the received image is badly affected by motion blur and signal interference during the transmission. However, the failed cases can be detected and discarded before the Kalman filter is updated. Only the estimated velocities between $v_{\min}$ and $v_{\max}$ are used as measurements for the Kalman filter. In our implementation, we set $v_{\min} = 10^{-6}$ m/s and $v_{\max} = 2$ m/s. While the filter is updated only with the valid measurements, it predicts the states at a constant frequency using the prediction model.

4 CONTROL SYSTEM DESIGN

Control methodology of the quadrotor MAV includes three major parts: inner-loop control, outer-loop control and reference generation, which is displayed in Fig. 6. The inner-loop control stabilizes the orientation of the system while the outer-loop control deals with the position, velocity and acceleration of the MAV in the north-east-down (NED) frame. The inner-loop dynamics, which is the part in the blue box in Fig. 6, will not be discussed here since in the previous work, the control strategies of orientation stabilization are investigated with the implementation of linear quadratic regulator (LQR) control law [19].

4.1 Position Control with Optical Flow

A position controller is designed and implemented with Robust and Perfect Tracking (RPT) method, which can be viewed in Fig. 6. As described in Fig. 2, the outer-loop control is realized by two parts: onboard height control with ultrasonic sensor measurement and GCS 2-D position control with optical flow based motion estimation. This control scheme is verified with a high-precision motion capture system VICON, which was discussed in [19] with flight test results.

In terms of RPT position control, the closed inner-loop can be treated as a virtual actuator (see [20]), the outer-loop

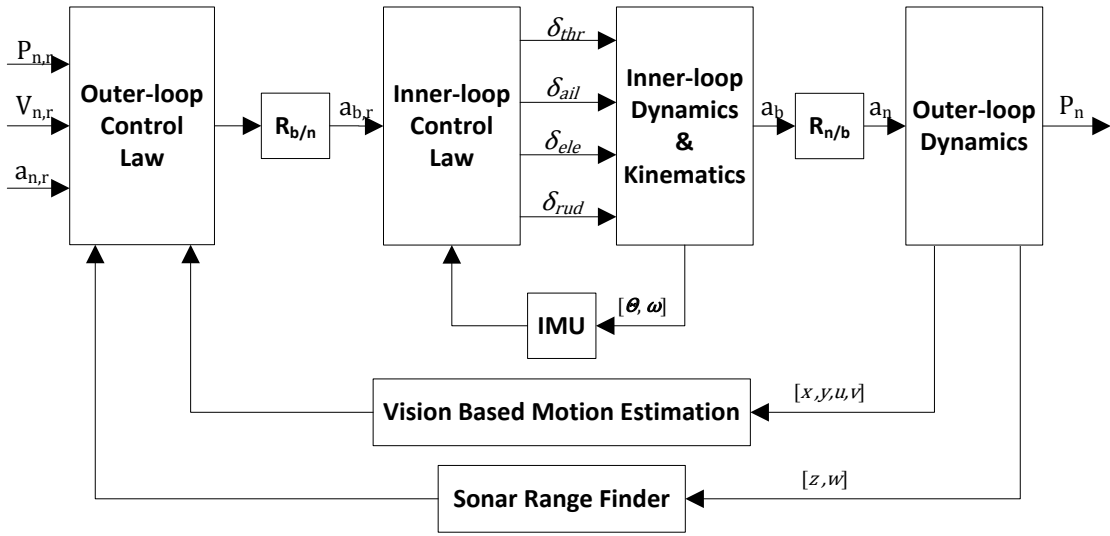


Figure 6: Structure of quadrotor MAV control system

dynamics of the aircraft can be regarded as the dynamics of a universal 6 degree-of-freedom (DoF) particle, without considering the coupling effect, which can be expressed as below:

$$\begin{pmatrix} \dot{x}_n \\ \dot{u}_n \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} x_n \\ u_n \end{pmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a_{x,n} \quad (13)$$

where $x_n, u_n, a_{x,n}$ are respectively position, velocity and acceleration for x directions in the NED frame. By applying the RPT approach introduced in [21], we can obtain an augmented system of the following form:

$$\Sigma_{\text{AUG}} : \begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w} \\ \mathbf{y} = \mathbf{x} \\ \mathbf{h} = \mathbf{C}_2\mathbf{x} \end{cases} \quad (14)$$

where

$$\mathbf{x} = \begin{pmatrix} \int x_e \\ x_{n,r} \\ u_{n,r} \\ a_{x,n,r} \\ x_n \\ u_n \end{pmatrix} \quad \mathbf{w} = \dot{a}_{x,n,r} \quad (15)$$

with $x_e = x_{n,r} - x_n$ as the position error and $\dot{a}_{x,n,r}$ as the derivative of the acceleration. By following the procedures in [22], a linear feedback control law can be formulated as:

$$\mathbf{u} = \mathbf{F}\mathbf{x}, \quad (16)$$

with

$$\mathbf{F} = \begin{pmatrix} \frac{k_i \omega_n^2}{\epsilon^3} \\ \frac{\omega_n^2 + 2\zeta \omega_n k_i}{\epsilon^2} \\ \frac{2\zeta \omega_n + k_i}{\epsilon} \\ 1 \\ -\frac{\omega_n^2 + 2\zeta \omega_n k_i}{\epsilon^2} \\ -\frac{2\zeta \omega_n + k_i}{\epsilon} \end{pmatrix}^T \quad (17)$$

where ϵ is the design parameter to adjust the settling time of the closed-loop system, ω_n, ζ and k_i are respectively the nominal natural frequency, damping ratio and desired pole location of the closed-loop system of the system (14),

$$p_i(s) = (s + k_i)(s^2 + 2\zeta \omega_n s + \omega_n^2) \quad (18)$$

The parameter ϵ is designed as a small number to achieve fast response. However, due to the limitation of MAV dynamics, the outer-loop bandwidth is chosen as smaller than 1/3 of the inner-loop bandwidth [23].

4.2 Path Generation

In Fig. 6, a smooth reference trajectory in NED frame inclusive of position $P_{n,r}$, velocity $V_{n,r}$ and acceleration $a_{n,r}$ is generated. In cooperation with the control and localization algorithm, a reference trajectory with continuous and limited velocity and acceleration is preferred. It prevents spikes in control input and helps the localization algorithm to achieve a stable and smooth performance. In this paper, a B-spline

based optimization algorithm is adopted to generate a C_2 continuous trajectory whose derivatives are well constrained. A similar method was adopted in the paper [24], except in this application we used one non-linear programming instead of the two layer of quadratic programming. A general form of B-spline is described in [25] as:

$$\begin{aligned} C(\mu) &= \sum_{i=0}^n N_{i,p}(\mu) P_i \\ \mu &= [\mu_0, \mu_1, \mu_2, \dots, \mu_n] \end{aligned} \quad (19)$$

where $C(\mu)$ denotes the reference trajectory, $N_{i,p}(\mu)$ is the basis function and P_i is the control point acquired from user input. The derivative of the B-spline is given as:

$$\begin{aligned} \frac{d}{d\mu} C(\mu) &= \sum_{i=0}^{n-1} N_{i+1,p-1}(\mu) Q_i \\ Q_i &= \frac{p}{\sum_{j=i+1}^{i+p+1} T_j} (P_{i+1} - P_i) \end{aligned} \quad (20)$$

where $T_j = \mu_j - \mu_{j-1}$ is the time segment and Q_i is the control point of the first order derivative. In order to obtain a time optimal trajectory, the overall time span of the trajectory is made as short as possible. Thus, the following optimization problem could be formulated:

$$\begin{aligned} \min \sum T_j^2 & \quad \text{subject to} \\ V_{\min} & \leq Q_i = \frac{p}{\sum_{j=i+1}^{i+p+1} T_j} (P_{i+1} - P_i) < V_{\max} \\ a_{\min} & \leq R_i = \frac{p-1}{\sum_{j=i+1}^{i+p+1} T_j} (Q_{i+1} - Q_i) < a_{\max} \end{aligned} \quad (21)$$

where R_i is the control point of the second derivative and $V_{\max}$, $V_{\min}$, $a_{\max}$, $a_{\min}$ are the upper and lower bounds for velocity and acceleration correspondingly. According to [26], the inequality constraints in (21) serve as the sufficient condition to limit the whole trajectory's velocity and acceleration within any user specified range. Problem in (21) could be solved easily using off-the-shelf non-linear programming package such as *fmincon* from Matlab.

5 EXPERIMENTAL RESULTS

In this section, flight tests are carried out to verify the control methods with our motion estimation algorithm. With the path generator mentioned in the last section, a path along a 2×2 m square was generated. Since the measurements of the ultrasonic sensor cannot be obtained at the GCS, the platform is maintained at 0.75 m height to provide the scale reference to the motion estimation. Fig. 7 shows the 3-D plot of the tracking performance. Fig. 8 presents the comparison between the measurements and references in x and y directions of the inertia frame respectively. To verify the performance of the height control, the measurements of the ultrasonic sensor is recorded onboard. The result is presented in Fig. 9. To further demonstrate the autonomy of our platform, a path along

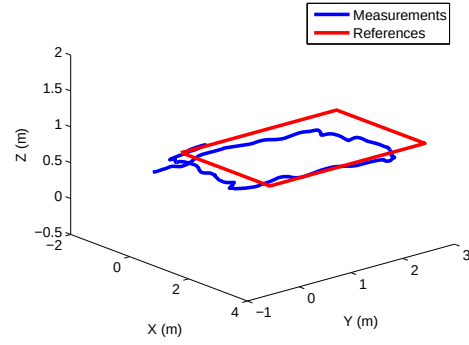


Figure 7: Autonomous flight of a square path

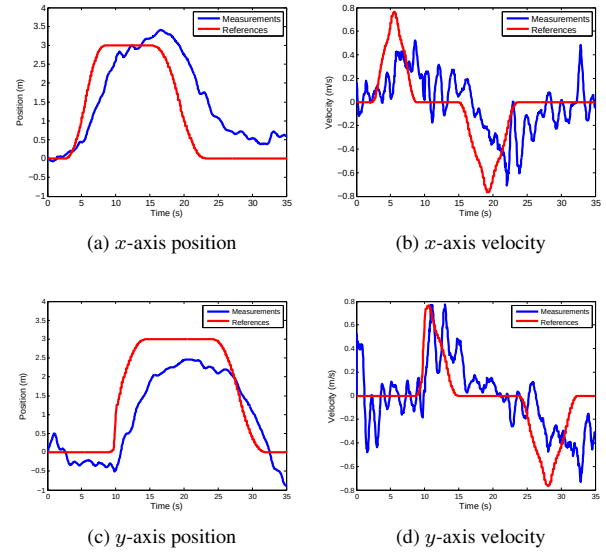


Figure 8: Tracking performance along x and y axes of the square path

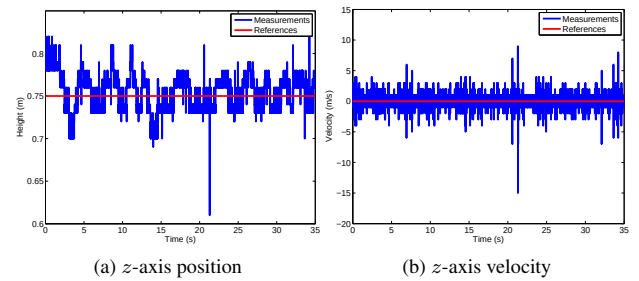


Figure 9: Tracking performance along z axis of the square path

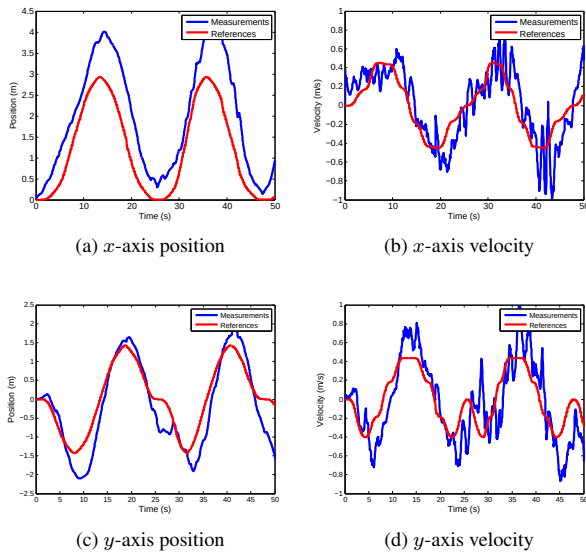


Figure 10: Tracking performance along x and y axes of the circular path

a circle with 2 m radius is then generated for the MAV to follow. It can be seen in Fig. 10, the MAV tracked the path well with no delays in both position and velocity. As there are several overshoot and drifting error in positions and oscillation in velocities for both square and circular paths, some possible causes are investigated as below:

1. Errors from scale factor. To accurately estimate the x and y axis velocity and position, the height of the MAV needs to be fixed at a certain altitude to obtain a precise scale factor. It can be seen in Fig. 9, the controlled height has noise and error up to 0.1 m compared with the set-point 0.75 m. As the height measurement from the ultrasonic sensor is not transmitted to GCS (no Wi-Fi module is developed for this platform considering the weight and computational load), the height channel errors are reflected by the errors of x and y axis due to the fixed scale in GCS. Furthermore, as the attitude measurement from IMU cannot be sent to GCS either, images are not rectified based on current attitude.
2. Severe noise in images due to signal interference causes error in the vision-based motion estimation. As stated in Fig. 5, images transmitted by analog transceiver are subject to interference and motion blur, which cause errors in feature matching for optical flow method. These problems become more severe when the transmission is blocked between the MAV and the receiver. Moreover, the optical flow method is subject to drifting error in position estimation even for clear and relatively high resolution images. Thus, it can be seen

that there are overshoot and drift error in both cases.

3. Errors from the platform. Fig. 8a and 8c shows a delay of response and an offset with respect to the reference around 2 seconds. Latency introduced by transmission from onboard analog camera to ground control station as well as image processing may cause delay in response of MAV. Further, the rotational rates of the brushed motors cannot be very precisely controlled by a chopper circuit ESC, oscillations in MAV motions are thus introduced. Also temperature rise may cause changes in the motor model, the MAV will have severe oscillations when operating for a certain period.

6 CONCLUSION

This paper presented the autonomous control design of a quadrotor MAV with regard to its hardware and software development. Based on the previous work of the ultra-light platform design as well as mathematical modeling and control, the platform equipped with a miniature ultrasonic range finder and a micro self-powered video system is capable of pre-planned trajectory tracking in an indoor environment based on visual odometry. A model based RPT control method is applied to global position control. In the current stage, we are still tuning the parameters of inner and outer loop control law to improve the control performance. In future, we aim at autonomous navigation of the MAV using 3-D vision based localisation and mapping algorithms.

REFERENCES

- [1] S. Shen, M. Nathan, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained mav. In *International Conference on Robotics and automation*, pages 20–25, Shanghai, China, 2011.
- [2] V. Lippiello, G. Loianno, and B. Siciliano. Mav indoor navigation based on a closed-form solution for absolute scale velocity estimation using optical flow and inertial data. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 3566–3571, Orlando, FL, USA, 2011.
- [3] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart. Mav navigation through indoor corridors using optical flow. In *IEEE International Conference on Robotics and Automation*, pages 3361–3368, Anchorage, AK, USA, 2010.
- [4] R. J. Wood, S. Avadhanula, M. Seeman and J. Entwistle E. Steltz, A. Bachrach, G. Barrows, and S. Sanders. An autonomous palm-sized gliding micro air vehicle. *Robotics & Automation Magazine*, 14(2):82–91, 2007.
- [5] F. M. Benjamin and R. J. Wood. Open-loop roll, pitch and yaw torques for a robotic bee. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 113–119, Vilamoura, Portugal, 2012.

- [6] J. J. Gibson. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, Boston, 1986.
- [7] W. Ding, J. Wang, S. Han, A. Almagbile, M. A. Garratt, A. Lambert, and J. J. Wang. Adding optical flow into the gps/ins integration for uav navigation. In *International Global Navigation Satellite Systems Society Symposium*, pages 1–13, 2009.
- [8] M. Achtelik, S. Weiss, and R. Siegwart. Onboard imu and monocular vision based control for mavs in unknown in-and outdoor environments. In *IEEE international Conference on Robotics and automation (ICRA)*, pages 3056–3063, Shanghai, China, 2011.
- [9] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys. Vision-based autonomous mapping and exploration using a quadrotor mav. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4557–4564, Vilamoura, Portugal, 2012.
- [10] F. Ruffier and N. Franceschini. Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 14, pages 2339–2346, New Orleans, USA, 2012.
- [11] G. de Croon, M. A. Groen, C. De Wagter, B. Remes, R. Ruijsink, and B. W. van Oudheusden. Design, aerodynamics and autonomy of the delfly. *Bioinspiration & biomimetics*, 7(2):1–36, 2012.
- [12] K. Li, S. K. Phang, B. M. Chen, and T. H. Lee. Platform design and mathematical modeling of an ultralight quadrotor micro aerial vehicle. In *International Conference on Unmanned Aircraft Systems*, pages 1077–1086, Atlanta, GA, USA, 2013.
- [13] G. Klein and D. W. Murray. Parallel tracking and mapping for small ar workspaces. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234, Nara, Japan, 2007.
- [14] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, WA, USA, 1994.
- [15] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, volume 81, pages 674–679, Vancouver, BC, Canada, 1981.
- [16] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision*. Springer, New York, NY, 2004.
- [17] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [18] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [19] S. K. Phang, K. Li, K. H. Yu, B. M. Chen, and T. H. Lee. Systematic design and implementation of a micro unmanned quadrotor system. *Unmanned Systems*, 2(2):121–141, 2014.
- [20] G. Cai, B. M. Chen, and T. H. Lee. *Unmanned Rotorcraft Systems*. Springer, New York, NY, 2011.
- [21] B. Wang, X. Dong, B. M. Chen, T. H. Lee, and S. K. Phang. Formation flight of unmanned rotorcraft based on robust and perfect tracking approach. In *American Control Conference*, pages 3284–3290, Montreal, QC, 2012.
- [22] B. M. Chen. *Robust and H_∞ Control*. Springer, New York, NY, 2000.
- [23] F. Wang, K. Wang, S. Lai, S. K. Phang, B. M. Chen, and T. H. Lee. An efficient uav navigation solution for confined but partially known indoor environments. In *11th IEEE International Conference on Control & Automation*, 2014 (to appear).
- [24] S. K. Phang, S. Lai, F. Wang, M. Lan, and B. M. Chen. UAV calligraphy. In *Proceedings of the 11th IEEE International Conference on Control & Automation*, pages 422–428, Taichung, Taiwan, 2014.
- [25] C. de Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, NY, 1978.
- [26] H. Kano, H. Fujioka, and C. F. Martin. Optimal smoothing and interpolating splines with constraints. *Applied Mathematics and Computation*, 5(218):1831–1844, 2001.

Actuator Fault Detection and Diagnosis for Quadrotors

Peng Lu^{*1}, Erik-Jan van Kampen¹, and Bin Yu²

¹Delft University of Technology, Delft, 2600 GB, The Netherlands

²Concordia University, Montreal, QC H3G 1M8, Canada

ABSTRACT

This paper presents a method for fault detection and diagnosis of actuator loss of effectiveness for a quadrotor helicopter. This paper not only considers the detection of the actuator loss of effectiveness faults, but also addresses the diagnosis of the faults. The detection and estimation of the faults are performed by the Augmented Extended Kalman Filter. The faults are modelled as random walk processes and are treated as additional states which makes the fault estimation unbiased. The estimated faults can be further used for Fault Tolerant Control. Simulation both without and with the occurrence of the actuator faults demonstrates the effectiveness of the proposed approach.

1 INTRODUCTION

Fault Detection and Diagnosis (FDD) is essential for achieving a successful control system reconfiguration [1]. Many techniques have been proposed especially for sensor and/or actuator faults [2, 3, 4]. Among these, model-based techniques have been widely studied and even successfully applied to several industrial areas [3, 5]. Many fault detection filters have been proposed to tackle this problem [3, 6].

Kalman Filter (KF) [7] is an optimal filter for linear system when the system is working nominally. If there are model uncertainties or faults, the state estimate of the filter will deviate from the real state. Furthermore, in practice, the system are always nonlinear if no assumptions are made. Therefore, the KF should be extended to nonlinear filter in order to cope with nonlinearities in the system. One way is to use the Extended Kalman Filter (EKF) [8], which requires the calculation of Jacobian matrix of the system matrix.

In this paper, the loss of effectiveness fault of a quadrotor helicopter is considered. In [9], the faults were detected by using the innovation of the KF. However, how to estimate the faults which are essential for Fault Tolerant Control (FTC) was not addressed. This paper uses the same fault detection technique and also addresses the fault diagnosis. The faults will be estimated by the Augmented Extended Kalman Filter (AEKF). The AEKF models the faults as additional states. The estimated faults can provide a more accurate information for the FTC.

^{*}Email address: P.Lu-1@tudelft.nl

The structure of the paper is as follows: Section 2 will present some preliminaries related to this paper. Section 3 will present the mathematical model of the quadrotor helicopter which is used in this paper. The modelling of the actuator faults is also introduced. The FDD of the actuator loss of effectiveness fault is presented in Section 4. The result without and with the fault are shown in Section 5 while Section 6 concludes the paper.

2 PRELIMINARIES

This section will present the problem which will be coped with in this paper.

2.1 Nonlinear system with input faults

Consider the nonlinear stochastic system with input faults in the discrete form

$$\begin{cases} x_{k+1} = f(x_k, u_k, k) + w_k + f_{i,k} \\ y_{k+1} = h(x_{k+1}, k+1) + v_{k+1} \end{cases} \quad (1)$$

where $x_{k+1} \in \mathbb{R}^n$ represents the system states, $u_k \in \mathbb{R}^m$ the control input, $y_{k+1} \in \mathbb{R}^p$ the measurement. w_k and v_k are the process noise and measurement noise vector respectively. The function $f_{i,k}$ represents input faults respectively. It is assumed that the noise vectors w_k and v_{k+1} are zero-mean and

$$E\{w_k\} = 0, \quad E\{w_k w_k^T\} = Q_k \delta_{k\tau} \quad (3)$$

$$E\{v_k\} = 0, \quad E\{v_k v_k^T\} = R_k \delta_{k\tau} \quad (4)$$

$$E\{w_k v_k^T\} = 0 \quad (5)$$

where the $\delta_{k\tau}$ denotes the Kronecker delta function, Q_k and R_k are the covariance matrix of the process noise and measurement noise respectively.

2.2 Extended Kalman Filter

This section presents the EKF [8, 10] which is used for the state estimation of the nonlinear system. The EKF, which is a form of the KF extended to nonlinear systems, can be applied to estimate the states. The five standard steps for an EKF are as follows:

1. One step ahead prediction

$$\hat{x}(k+1|k) = \hat{x}(k|k) + \int_k^{k+1} f(x(t), u(t), t) dt \quad (6)$$

2. Covariance matrix of the state prediction error

$$P(k+1|k) = \Phi(k, \tau)P(k|k)\Phi^T(k, \tau) + Q_d(k) \quad (7)$$

where $\Phi(k, \tau)$ is calculated as follows:

$$\Phi(k, \tau) = e^{F(k)\Delta t} = \sum_n \frac{F_k^n (\Delta t)^n}{n!} \quad (8)$$

where $F(k)$ is the linearized matrix computed as follows:

$$F(k) = \left. \frac{\partial f(x(t), u(t), t)}{\partial x(t)} \right|_{x=\hat{x}(k|k)} \quad (9)$$

The $Q_d(k)$ is calculated as follows:

$$\begin{aligned} Q_d(k+1|k) &= E\{w_d(k)w_d^T(k)\} \\ &= \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)G_k Q_k G_k^T \Phi^T(t_{k+1}, \tau) d\tau \\ &\approx \sum_m \sum_n \frac{F_k^m}{m!} G_k Q_k G_k^T \frac{F_k^n}{n!} \frac{(\Delta t)^{n+m+1}}{(n+m+1)!} \quad (10) \end{aligned}$$

However, there is a commonly used approximation for $Q_d(k)$ which is calculated as follows:

$$Q_d(k) = \Gamma(k)Q(k)\Gamma(k)^T \quad (11)$$

where $\Gamma(k)$ is calculated as follows:

$$\Gamma(k) = \left(\int_{k-1}^k \Phi(k) \Delta t \right) G(k) \quad (12)$$

3. Compute the Kalman gain

The optimal Kalman gain is calculated by the following equation:

$$K(k+1) = P(k+1|k)H^T(k+1)V^{-1}(k+1) \quad (13)$$

where $V(k+1)$ is calculated by

$$V(k+1) = H(k+1)P(k+1|k)H^T(k+1) + R(k+1) \quad (14)$$

where $H(k+1)$ is the linearized matrix of the measurement matrix

$$H(k) = \left. \frac{\partial h(x(t), u(t), t)}{\partial x(t)} \right|_{x=\hat{x}(k|k)} \quad (15)$$

4. Measurement update step:

$$\begin{aligned} \hat{x}(k+1|k+1) &= \hat{x}(k+1|k) \\ &\quad + K(k+1)\gamma(k+1) \quad (16) \end{aligned}$$

where $\gamma(k)$ is the innovation of the EKF, which is computed as

$$\gamma(k+1) = (y(k+1) - H(k+1)\hat{x}(k+1|k)) \quad (17)$$

5. Update the covariance matrix of the state estimation error matrix

$$\begin{aligned} P(k+1|k+1) &= (I - K(k+1)H(k+1))P(k+1|k) \\ &\quad \times (I - K(k+1)H(k+1))^T \\ &\quad + K(k+1)R(k+1)K^T(k+1) \quad (18) \end{aligned}$$

Since the EKF is a linearized form of the KF, problems may occur when the linearisation error is large. Note the EKF is only convergent when the initial states are close enough to the real states.

3 MATHEMATICAL MODEL AND FAULT MODEL OF THE QUADROTOR

This section will present the mathematical model of the quadrotor and build the fault model for the actuator faults of this quadrotor. Note the model used here is the same with [9]. The model used is the Qball-X4 model.

3.1 Model of the quadrotor

In this paper, the objective is to detect and estimate the faults of the quadrotor actuators. The model of a quadrotor can be described as follows [11]:

$$\dot{x} = v_x \quad (19)$$

$$\dot{y} = v_y \quad (20)$$

$$\dot{z} = v_z \quad (21)$$

$$m\dot{v}_x = U(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \quad (22)$$

$$m\dot{v}_y = U(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \quad (23)$$

$$m\dot{v}_z = U(\cos \phi \cos \theta) - mg \quad (24)$$

$$J_1 \ddot{\phi} = \tau_\phi \quad (25)$$

$$J_2 \ddot{\theta} = \tau_\theta \quad (26)$$

$$J_3 \ddot{\psi} = \tau_\psi \quad (27)$$

where x , y and z are the positions of the quadrotor, v_x , v_y and v_z are the velocity components of the quadrotor. m is the mass of the quadrotor and g is the gravity constant. ϕ , θ and ψ are the attitude angles of the quadrotor which are roll, pitch and yaw angle respectively. J_1 , J_2 and J_3 are the moments of inertia of the quadrotor. U is the total lift and τ_ϕ , τ_θ and τ_ψ are the torques along the directions of the ϕ , θ and ψ angles respectively.

It is assumed that all the states mentioned above are measured. However, the measurements are corrupted by noise. The measurement model is as follows:

$$x_m = x + \nu_x \quad (28)$$

$$y_m = y + \nu_y \quad (29)$$

$$z_m = z + \nu_z \quad (30)$$

$$v_{xm} = v_x + \nu_{vx} \quad (31)$$

$$v_{ym} = v_y + \nu_{vy} \quad (32)$$

$$v_{zm} = v_z + \nu_{vz} \quad (33)$$

$$\phi_m = \phi + \nu_\phi \quad (34)$$

$$\theta_m = \theta + \nu_\theta \quad (35)$$

$$\psi_m = \psi + \nu_\psi \quad (36)$$

where $\nu = [\nu_x \ \nu_y \ \nu_z \ \nu_{vx} \ \nu_{vy} \ \nu_{vz} \ \nu_\phi \ \nu_\theta \ \nu_\psi]^T$ are the noises in the measurements.

3.2 Model of the actuator

This subsection will introduce the model of the actuator and the fault modelling of the actuator faults. The actuator dynamics are as follows:

$$\dot{U} = \frac{K\omega}{s + \omega} u \quad (37)$$

where K is a gain factor and ω is a parameter of the actuator. The input of the actuator is the PWM. Therefore, the model of the actuator with the PWM input is as follows:

$$\dot{U} = -\omega U + K\omega \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} PWM_{1t} \\ PWM_{2t} \\ PWM_{3t} \\ PWM_{4t} \end{bmatrix} \quad (38)$$

where PWM_{it} , $i = 1, 2, 3, 4$ are the theoretical four PWM inputs respectively. Also note that the initial PWM outputs are the minimum throttle value 0.05 [9]. Therefore, the PWM_{it} should be changed to

$$PWM_{it} = PWM_{ir} - 0.05, \quad i = 1, 2, 3, 4 \quad (39)$$

where $PWM_{ir} \in (0.05, 0.1)$ are the real PWM inputs to the actuator.

3.3 Actuator fault modelling

This subsection will consider the modelling of the actuator faults of the quadrotor. The fault considered in this paper is partial loss of control effectiveness. Let l_i , $i = 1, 2, 3, 4$ denote the loss of effectiveness. Then the model of the actuator

is

$$\begin{aligned} \dot{U} &= -\omega U + K\omega \begin{bmatrix} 1 - l_1 & 1 - l_2 & 1 - l_3 & 1 - l_4 \end{bmatrix} \begin{bmatrix} PWM_{1t} \\ PWM_{2t} \\ PWM_{3t} \\ PWM_{4t} \end{bmatrix} \\ &= -\omega U + K\omega \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} PWM_{1t} \\ PWM_{2t} \\ PWM_{3t} \\ PWM_{4t} \end{bmatrix} \\ &\quad - K\omega [PWM_{1r}, PWM_{2r}, PWM_{3r}, PWM_{4r}] \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \end{bmatrix} \end{aligned} \quad (40)$$

Now the actuator fault model has been established, the objective of this paper is to detect and diagnose the faults, which will be introduced in the following section.

4 FAULT DETECTION AND DIAGNOSIS OF THE ACTUATOR FAULTS

This section will present the method for the detection and diagnosis of the loss of effectiveness fault for the quadrotor. In [9], the actuator faults can be detected using the innovation of the EKF. However, the fault reconstruction is not considered which is essential for fault tolerant control. The fault estimation is also beneficial for the diagnosis of the system. In this section, the fault detection will be first introduced.

The fault detection is based on the innovation of the filter

$$\gamma(k+1) = (H(k+1)x(k+1) - H(k+1)\hat{x}(k+1|k)) \quad (41)$$

In order to detect the faults, a threshold T_0 needs to be defined. The fault detection logic is

$$\gamma(k+1) = \begin{cases} \geq T_0 \Rightarrow \text{trigger the fault alarm} \\ < T_0 \Rightarrow \text{No faults} \end{cases} \quad (42)$$

Since there are noises in the measurement, T_0 should be a number bigger than zero. One can also accumulate the innovation in a certain length of time window to detect the faults.

4.1 Augmented Extended Kalman Filter

In this paper, we propose to use the AEKF to estimate the faults in the actuators of the quadrotor. The AEKF is an augmented form of the EKF. The faults are augmented as additional states so the faults are also estimated by the AEKF.

To estimate the faults, we need to have a fault model. One way to model the dynamics of the time-varying input fault is to treat it as a random walk process[12, 13, 14]. This strategy is combined with the filter for the purpose of improving the state estimation performance in the presence of unknown input faults. In order to achieve this, a random signal w'_k with a

covariance matrix Q'_k is introduced to the dynamic model of the filter

$$\dot{x}'(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{f} \end{bmatrix} = \begin{bmatrix} f(x(t), u(t), t) \\ 0 \end{bmatrix} + \begin{bmatrix} w(t) \\ w'_k(t) \end{bmatrix} \quad (43)$$

where f is the vector for the loss of effectiveness fault. Since there are four PWM inputs, the vector of the faults is denoted as

$$f = [f_1 \ f_2 \ f_3 \ f_4]^T \quad (44)$$

By augmenting the faults as the state vector, the faults can be estimated in an unbiased sense. The value of the f_i , $i = 1, 2, 3, 4$ is the estimated value of l_i . Therefore, the estimation of the loss of effectiveness fault is achieved.

5 SIMULATION RESULTS

In this section, the approach proposed in the previous section will be applied to the quadrotor actuator fault diagnosis. The baseline controller is a PID controller which controls the altitude of the quadrotor. The position x and y are not controlled.

In order to show the performance of the approach, the situation with and without actuator faults will be shown.

5.1 FDD when there are no faults

In this simulation, there are no actuator faults. All the PWM inputs are normal. This means that $l_1 = l_2 = l_3 = l_4 = 0$. The results of the actuator fault diagnosis using the AEKF are shown in Figure 1 - Figure 5.

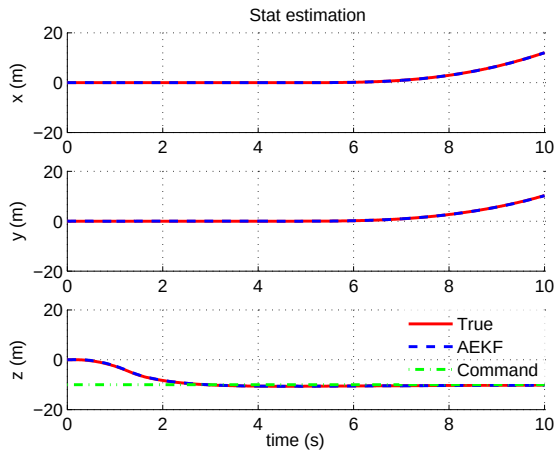


Figure 1: State estimation of the KF without faults

Figure 1 shows the controller response. The controller can track the command of the altitude which is given in green line. The red lines denote the true states of the quadrotor while the blue dotted lines represent the estimation of the AEKF. Since there is only a controller for the altitude, the position x and y are not controlled.

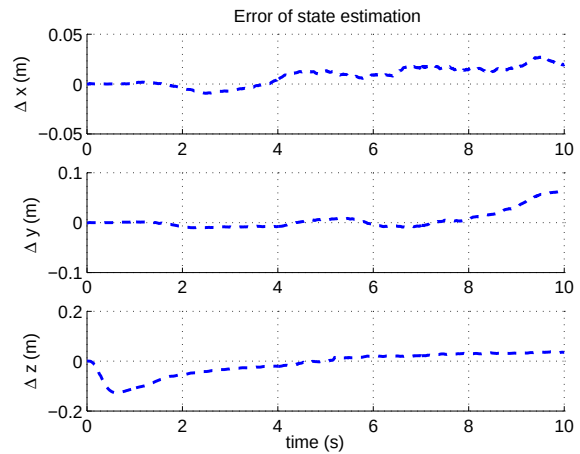


Figure 2: State estimation of the KF without faults

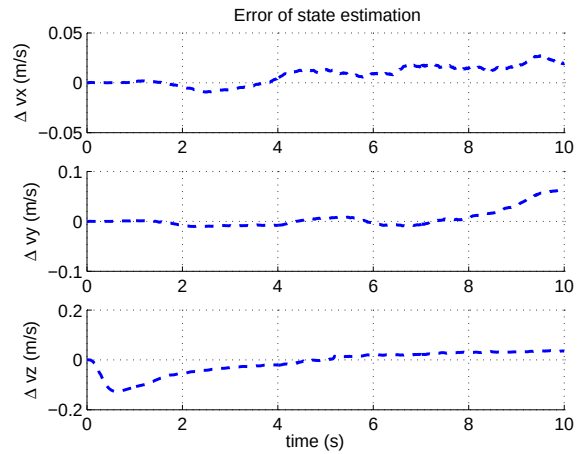


Figure 3: State estimation of the AKF without faults

The state estimation error of the position vector, velocity vector and attitude angles using the AEKF are shown in Figure 2, Figure 3 and Figure 4 respectively. As can be seen, all the estimation error are close to zero-mean. This demonstrates the state estimation ability of the AEKF.

The fault estimation using the AEKF is shown in Figure 5. As can be seen, the estimation requires almost one seconds to converge to the true value. After that, the estimation error is zero-mean.

5.2 FDD when there are actuator faults

In this simulation, all the four PWM suffer from a loss of effectiveness fault. The magnitude of the fault is $0.1t$, and the limit for the loss of effectiveness factor is 0.5. The faults of two actuators related to the pitch axis are injected at $t = 5$ s whereas those of two actuators related to the roll axis are injected at $t = 6$ s.

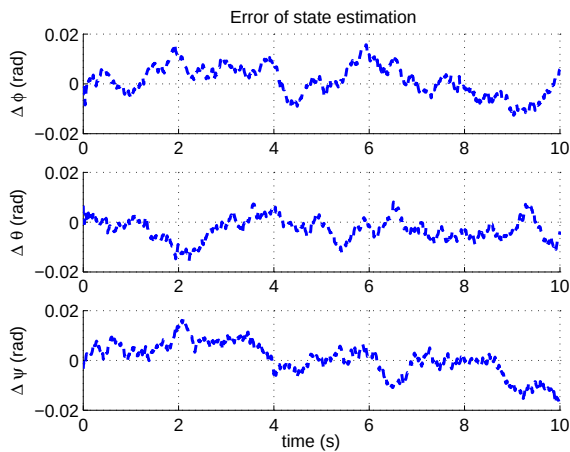


Figure 4: State estimation of the AKF without faults

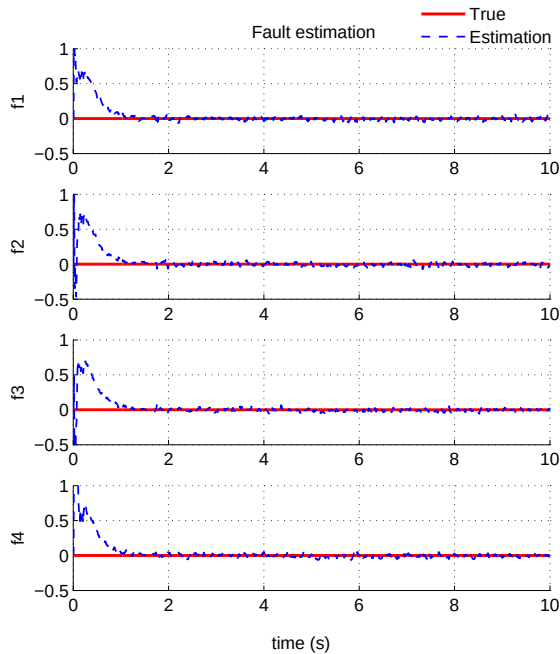


Figure 5: Fault estimation of the AKF without faults

Note the faults injected here shows a loss of effectiveness fault. The result is shown in Figure 6.

The state estimation errors using the AEKF are similar to the situation when there are no actuator faults. Therefore, they are not shown here. The controller response is also similar to Figure 1. The fault estimation using the AEKF is given in Figure 6. As can be seen from the figure, the loss of effectiveness faults of the four actuators can be estimated correctly. However, there is a time delay between the true faults and the estimated faults. This can be improved by changing the mag-

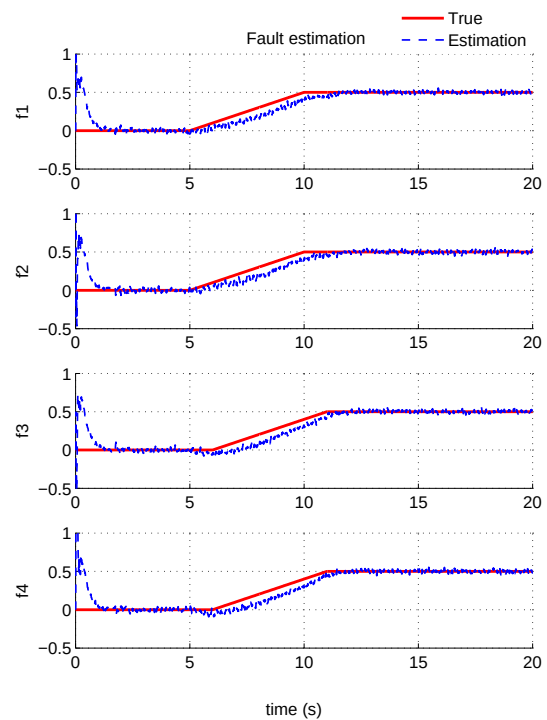


Figure 6: State estimation of the KF with faults

nitude of the random signal $w'(k)$ in Equation 43. Increasing the magnitude can reduce the time delay. However, the noise magnitude of the fault estimation will be bigger.

6 CONCLUSIONS

This paper addresses the FDD of the actuator loss of effectiveness fault for a quadrotor helicopter. The model of the quadrotor was built and the modeling of the faults was also introduced. The detection of the faults was addressed by the innovation of the filter. The diagnosis of the faults was addressed by modelling the faults as a random walk process. Therefore, the estimation of the fault achieved an unbiased estimate by treating the faults as additional states. The estimated faults can be used for FTC. Two situation was simulated to show the performance of the proposed approach. The controller was able to follow the command successfully both with and without the faults. The faults were also estimated in an unbiased sense, which verified the effectiveness of the approach.

Future work includes designing a controller for the position including x and y . Furthermore, the implementation of the proposed approach on a real quadrotor should also be carried out.

REFERENCES

- [1] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual*

- Reviews in Control*, 32:229–252, 2008.
- [2] Rolf Isermann. Model-based fault-detection and diagnosis status and applications. *Annual Reviews in Control*, 29(1):71–85, January 2005.
 - [3] Christopher Edwards, Sarah K. Spurgeon, and Ron J. Patton. Sliding mode observers for fault detection and isolation. *Automatica*, 36:541–553, 2000.
 - [4] J. Marzat, H. Piet-Lahanier, F. Damongeot, and E. Walter. Model-based fault diagnosis for aerospace systems: a survey. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 226(10):1329–1360, January 2012.
 - [5] Denis Berdjag, J. Cieslak, and Ali Zolghadri. Fault diagnosis and monitoring of oscillatory failure case in aircraft inertial system. *Control Engineering Practice*, 20:1410–1425, 2012.
 - [6] Ron J. Patton. Fault-tolerant Control Systems: The 1997 Situation. In *Proc. of IFAC Symp. on Fault Detection, Supervision and Safety for Technical Processes*, pages 1033–1054, 1997.
 - [7] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems 1. *Journal of Basic Engineering*, 82(Series D):35–45, 1960.
 - [8] Konrad Reif, S. Gunther, Engin Yaz, and Rolf Unbehauen. Stochastic Stability of the Discrete-Time Extended Kalman Filter. *IEEE Transactions on Automatic Control*, 44(4):714–728, 1999.
 - [9] Bin Yu, Youmin Zhang, Yingmin Yi, Yaohong Qu, and Peng Lu. Fault detection for partial loss of effectiveness faults of actuators in a quadrotor unmanned helicopter. In *The 11th World Congress on Intelligent Control and Automation*, page (to appear), 2014.
 - [10] Thomas Lombaerts, Eddy Van Oort, Q.P. Chu, J. A. Mulder, and Diederick Joosten. Online Aerodynamic Model Structure Selection and Parameter Estimation for Fault Tolerant Control. *Journal of Guidance, Control, and Dynamics*, 33(3):707–723, May 2010.
 - [11] Rong Xu and Umit Ozguner. Sliding Mode Control of a Quadrotor Helicopter. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 4957–4962. Ieee, 2006.
 - [12] P. S. Maybeck. *Stochastic Models, Estimation, and Control*. Academic Press Inc., London, vol. 1 edition, 1979.
 - [13] J. A. Mulder, Q. P. Chu, J. K. Sridhar, J. H. Bree-man, and M. Laban. Non-linear aircraft flight path reconstruction review and new advances. *Progress in Aerospace Sciences*, 35:673–726, 1999.
 - [14] P. Lu, L. Van Eykeren, E. van Kampen, Q. P. Chu, and B. Yu. Adaptive Hybrid Unscented Kalman Filter for Aircraft Sensor Fault Detection, Isolation and Reconstruction. In *AIAA Guidance, Navigation, and Control Conference*, number January, pages 1–18, National Harbor, Maryland, 2014.

eMotionSpheres: A New Design for Human-Friendly UAVs

N. Gaissert^{1*}, R. Mugrauer², G. Mugrauer³, A. Jebens⁴, K. Jebens⁵, C. Rabe⁶, and E.M. Knubben⁷
Bionic Learning Network, Festo AG & Co. KG, Ruiter Strasse 82, 73734 Esslingen, Germany

ABSTRACT

Here we present a new design for human-friendly UAVs (Unmanned Aerial Vehicles). Contrary to conventional UAVs like gyrocopters or quadcopters, the so called eMotionSpheres feature high process stability, long endurance and furthermore allow for a safe human-machine interaction. Since these features are essential for UAVs used in a production environment the eMotionSpheres will open up new applications of UAVs in the future. Each flying object consists of a helium filled ballonnet and is driven by eight small propellers, which are attached to the outer shell. The propellers are a novelty when it comes to flying objects. They consist of a laser-sintered rigid outer frame which is spanned by a thin foil. This design, inspired by the dragonfly wing, results in a light-weight drive, which is adaptive and most interestingly supplies the same efficient thrust in a forward and a reverse direction. One or several of the eMotionSpheres can move in an autonomous or coordinated manner within a defined space. Therefore the flying objects communicate via active markers with a camera system. The cameras reconstruct a 3D space and together with a central master computer specify the flight paths of the spheres. Since the camera system is easy to install and calibrated in few minutes it offers, together with, but also without the spheres, several interesting applications in guidance and monitoring of the networked factory of the future.

1 INTRODUCTION

The civilian use of quadcopters and other Unmanned Aerial Vehicles (UAVs) grows steadily. Geologists use them to collect data about ancient ruins [1], biologists survey and map forests and biodiversity to protect animals and their natural habitat [2]. Mountain Rescue uses quadcopter to detect avalanche victims [3] and photographers and videographers provide totally new and impressive sights of sporting events. However, the industrial use of UAVs is still limited. The main reasons for this are the limitation in flight time, the process

stability that is costly to achieve and the risk of injuring humans in the case of failure.

Here we present a totally new design for human-friendly UAVs that, combined with an interactive infrared technology, provides high process stability, long endurance and allows for a safe human-machine interaction. Each sphere of the eMotionSpheres has a diameter of 95 cm and is filled with helium. The helium provides buoyancy to lift the spheres even when the battery runs empty.

Eight small propellers are attached to the outer shell of the spheres. They enable the spheres to fly, climb or descend quickly up and down and to move horizontally in any spatial direction to an accuracy of 1 cm. The propellers are inspired by the dragonfly wing and hence are very lightweight and flexible and thus are not dangerous to humans.

Ten infrared cameras installed in the room record the spheres via their active infrared markers. Compared to passive markers, a larger space can therefore be recorded with the same number of cameras. Furthermore the markers are not permanently lit, but instead flash for only 250 μ s which make the system highly energy-efficient and allows for long flight-times. The position data is passed on to a main computer. The actions calculated on the main computer are sent back to the objects, where they are locally implemented. On the computer there are preprogrammed paths, which specify the spheres flight paths when flying in formation. Thanks to additionally stored behaviour patterns, the spheres can however also move autonomously through the space. There are no collisions even in chaotic situations as they move out of each others way. As the spheres stop at their charging stations regularly and autonomously, they can be used as flying objects for several days without a person having to intervene.

2 FLYING OBJECTS WITH ADAPTIVE BIO-INSPIRED PROPELLERS

For the eMotionSpheres we combined knowledge obtained from rotorcrafts with knowledge obtained from aerostats. Rotorcrafts as helicopters or quadcopters are heavier than air and use lift generated by rotor blades while common aerostats are lighter than air. In balloon flight, buoyancy is provided by a carrier gas, e.g. hydrogen, helium, superheated steam or hot air, in an enclosed space. Since hydrogen is highly flammable it is not suitable to be used in a production environment. Steam or hot air are no viable alternatives either for long term use because it loses its buoyancy when cooling down. Helium is not used often since it is too

*Email address: niga@de.festo.com



Figure 1: eMotionSpheres: an interactive infrared camera system enables both the formation and individual flight of the spheres.

expensive. Here, however, we use a 7-layer bidirectionally stretched PE/PP film with a thickness of only $25\ \mu\text{m}$ and a weight of only $27\ \text{g/m}^2$. This foil is so dense that virtually no air leaks from the spheres allowing to generate a stable system for several days. This way the high price of the helium becomes reasonable.

Each sphere has a diameter of 95 cm. Since $1\ \text{m}^3$ of helium generates approximately 1.1 kg weight of buoyant force the helium can carry a weight of up to 500 g. Thereof the weight of the shell, the onboard electronics with 12 processors, the radio unit, the battery and eight LEDs, four infrared LEDs for communication with the cameras and four power LEDs for light signals, and especially the 8 propellers, have to be subtracted. The helium thus ensures that the spheres swim in a sea of air. This way the propellers do not have to carry the full weight of the spheres but are mainly used for the precise positioning of the spheres in 3D space. This setup results in an energy-efficient overall system since the propellers are only active when needed.

Eight propellers are attached along the equator of the spheres, four of them aligned horizontally. They enable the spheres to fly, climb or descend quickly up and down. Thanks to the four vertical propellers, the spheres can move horizontally in all directions and rotate about their vertical axis. Four of the propellers are screwed anticlockwise and the other four clockwise, with an anticlockwise drive always being located opposite a clockwise one. In this way, the torques are neutralised and the spheres can be steered precisely. The actions from the main computer are sent to the spheres and locally implemented by the activation of the eight brushless motors. The corresponding 8 processors are installed onboard the spheres.

The propellers represent something totally new when it comes to flying objects. For precise positioning of the spheres it was mandatory to have powerful rotors. However, they had

to be lightweight at the same time. Therefore we looked at an interesting natural role model: the dragonfly. Dragonflies execute the most complicated flight manoeuvres. Astonishingly, a wing weighs only about 0.002 g although they span a distance of up to 10 cm [4]. Dragonflies can reach a maximum speed of 54 km/h and with some tailwind they can fly up to 1000 km [5]. Some dragonflies even have been observed to fly with half a wing missing. The wings consist of a thin transparent membrane strengthened by a number of longitudinal veins. The membrane has a thickness of $3\ \mu\text{m}$ at its thinnest point [4]. The front edge of the wing consists of much thicker veins with a thickness of up to $220\ \mu\text{m}$. Thus the membrane is quite flexible compared to the rigid and stable edge. Based on this structure the dragonfly wings show an interesting effect. While flapping up and down the rigid edge leads the movement but the membrane moves like the

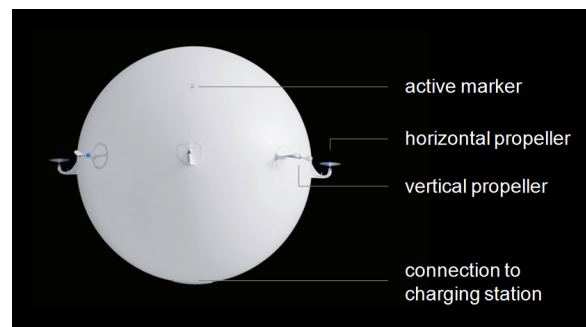


Figure 2: Each sphere has four vertical and four horizontal adaptive propellers. The four active markers are embedded into the shell. At the bottom of the sphere is a connection to the charging station. For charging, the sphere autonomously approaches the charging station then performs a 90° rotation to interlock with the station using a bayonet fitting.

sail of a sailing boat and thus form a profile comparable to the shape of rotor blades [6]. We analyzed the wing deformation of dragonfly wings when we were developing the BionicOpter, a Micro Aerial Vehicle, inspired by the dynamics of dragonfly flight [7]. The knowledge gained from the work on the dragonfly wings went into the design of the adaptive propellers.

Just like the wings of the dragonfly the spheres propellers are made of a sturdy frame covered with a flexible membrane. The frame is twisted once, thus making a figure of eight. The frame is made of laser-sintered polyamide and a few millimetres thick. The membrane consists of translucent PE and is only 5 μm thick. As the film is not under complete tension, it inflates on one side or the other, depending on which direction the propeller is turning. Thus the adaptive propellers have a profile modification based on the direction of rotation. With 7.8 V and 2.2 A they supply 42 g of thrust both forwards and backwards and are thus equally efficient in both directions. In flying objects, this efficient combination of equal thrust performance in two directions is a genuine first. Furthermore, it facilitates to balance the forces across the eight propellers of the spheres leading to the precise positioning performance of the eMotionSpheres. Finally the propeller weights less than 1 g. Due to this lightweight construction the rotation direction can be changed almost without any delay.

The lightweight design of the adaptive propellers has one additional important advantage to standard quadcopter or gyrocopter propellers: the propellers are flexible and lightweight and thus do not have to be shielded. The new propeller design, together with the buoyancy provided by the helium guarantees safe interaction between the spheres and humans.

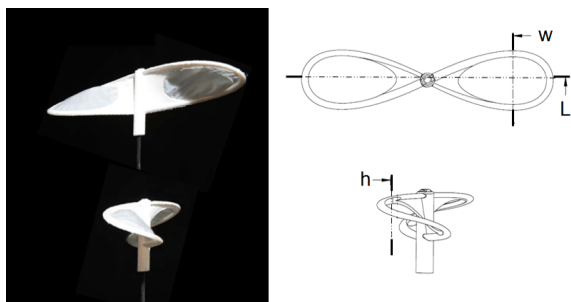


Figure 3: The adaptive propellers have a length of 9 cm, a width of 2.2 cm and a height of 1 cm. They consist of a laser-sintered sturdy frame covered with a flexible membrane, which is not under complete tension. Because of this flexible structure inspired by the dragonfly wing they are not dangerous for humans. Most interestingly the propellers generate identical thrust in both directions.

3 EXACT POSITIONING THANKS TO INTERACTIVE INFRARED TECHNOLOGY

The drives are adaptive and supply the same efficient thrust in both a forward and reverse direction. Together with the camera system and an intelligent control system, the eMotionSpheres are capable of the widest range of flight manoeuvres with or without a human pilot.

Most state-of-the-art infrared marker-tracking systems measure the displacement of objects by tracking a set of reflective dots mounted to the objects. Infrared cameras installed above the recording area are equipped with an array of infrared LEDs and send out light. This light is then reflected by the passive markers. The cameras then record a picture detecting the reflection. This way the light has to travel the distance from the camera and back. Because of this and because of the dispersion of reflection the size of the recorded space is limited.

For our system we use active markers instead. Four infrared LEDs are embedded in the shell of the spheres on the upper hemisphere of about 45° latitude. Compared to passive markers, a larger space can therefore be recorded with the same number of cameras. With the setup described here a space of more than 30 m to 30 m can easily be covered. The flashing of the active markers is not only used to define the position of the sphere in space but to identify its orientation as well. Furthermore, the pattern of light flashes is used to tell the optically identical flying objects apart.

The exact positioning is based on an interactive infrared technology. This means, that the cameras only take pictures when the active markers are illuminated. POINT GREY infrared cameras are positioned so that they map out the flight area as a whole. Due to special filters, the used cameras only capture infrared light and are not sensitive to other surrounding light. Because of the positioning of the active markers we installed the cameras in such a way that the cameras look down to the spheres from above. Each sphere is recorded by at least two cameras. Since we used 8 spheres for testing 10 cameras were installed. This way occlusions were ruled out efficiently. In order to know where the flying spheres are located in space, the computer first must know the positions of the cameras. Therefore a manually controlled sphere with a measuring cross of defined size flies through the space for 15 min while being recorded by all cameras. The recorded random trajectories are then used to recalculate the exact but relative positions and alignment of the cameras (for more information on multi camera calibration see e.g. [8]). In total the whole calibration process only takes a couple of minutes. Due to the relatively cheap cameras and because of the quick and easy calibration process the system is suitable to be used in a production environment.

Active markers however have one drawback which is the power consumption. To keep the power consumption of the four infrared LEDs to a minimum the markers are not permanently lit, but instead flash for only 250 μs . As a result the



Figure 4: Infrared cameras are used for exact positioning of the spheres.

markers are highly energy efficient. The short illumination times of the active markers makes it necessary to synchronize the cameras with the active markers. A newly designed board on the spheres communicates with a second board mounted to the cameras using radio signals.

In total 12 microprocessors are integrated into each sphere. 8 control the brushless motors of the rotors. The others are used for the interactive communication between the spheres and the cameras and for the communication with the main computer. The main computer defines the trajectories of the spheres while the stabilization of the spheres is calculated on board based on an 10 DOF IMU with integrated gyroscopes and acceleration sensors.

Equipped with a 2200 mAh Li-poly battery the average flight time of a sphere comes to two hours depending on the flight speed and manoeuvres that are flown. As the spheres stop at their charging stations regularly and autonomously, they can be used as flying objects for several days without a person having to intervene.

4 CONCLUSION

The eMotionSpheres represent a new design for human-friendly UAVs. The spheres are quickly moved in space using 8 adaptive lightweight rotors inspired by the dragonfly wings. The flexible propellers are extremely efficient in both directions and most importantly not dangerous to humans. Helium together with an autonomous charging system allows for flight time of several days. The interactive infrared technology allows to position the spheres precisely in an enlarged 3D space. Because of these reasons the eMotionSpheres open up new applications of UAVs in a production environment.

Generally speaking, the spheres could be conceivably used as a guidance system whether for people working in factories or for visitors at trade fairs, museums or large indoor events. Most interesting is the field of moving sensors [9]. If data has to be obtained from different points within a

production line the same sensor has to be installed at all of these points of interest. Some sensors however often must be calibrated or cleaned, or are just expensive. In such a case it makes sense to use one sensor and move it from point to point. In such a case the sensor could be mounted to a sphere which patrols the points of interest on a regular basis, collects the data and returns the sensor for calibration or cleaning. The camera system itself, with or without the spheres, could be used as a monitoring system of a factory. It pinpoints the positions of products in the factory, analyses them and thereby improves the production process. In the field of smart logistics [10] the system could be put to use as a way of monitoring the function of small robots or forklifts. Our system has a clear advantage here because of the enlarged space coverage resulting from the active markers. Finally, the system can also be used to improve human-machine interaction and increase the safety for workers. The system can track movements of workers within a factory and stop machines when the workers get too close to dangerous setups. Taken together the eMotionSpheres together with the interactive infrared camera system describes several new ideas that will open up applications of UAVs also in a production environment.

ACKNOWLEDGEMENTS

Several people helped with generating pictures, animations and videos. The authors want to thank all of these helpful colleagues.

REFERENCES

- [1] C Seitz and H Altenbach. Project archeye—the quadcopter as the archaeologists eye. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(1):C22, 2011.
- [2] LP Koh and SA Wich. Dawn of drone ecology: low-cost autonomous aerial vehicles for conservation. *Tropical Conservation Science*, 5(2), 2012.
- [3] M Meiboom, F Andert, S Batzdorfer, H Schulz, W Inninger, and A Rieser. Untersuchungen zum einsatz von uavs bei der lawinenrettung. In *Deutscher Luft- und Raumfahrt Kongress (DLRK) 2013*, 2013.
- [4] HX Zhao, YJ Yin, and Z Zhong. Assembly modes of dragonfly wings. *Microscopy research and technique*, 74(12):1134–1138, 2011.
- [5] J Silsby. *Dragonflies of the World*. CSIRO PUBLISHING, 2001.
- [6] SR Jongerius and D Lentink. Structural analysis of a dragonfly wing. *Experimental Mechanics*, 50(9):1323–1334, 2010.
- [7] N Gaissert, R Mugrauer, G Mugrauer, A Jebens, K Jebens, and EM Knubben. Inventing a micro aerial vehicle inspired by the mechanics of dragonfly flight. In

TAROS 2013: 14th Towards Autonomous Robotic Systems, 2013.

- [8] B Li, L Heng, K Koser, and M Pollefeys. A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1301–1307. IEEE, 2013.
- [9] B Liu, P Brass, O Dousse, P Nain, and D Towsley. Mobility improves coverage of sensor networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 300–308. ACM, 2005.
- [10] D Uckelmann. A definition approach to smart logistics. In *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, pages 273–284. Springer, 2008.

APPENDIX A:

A video showing the flight behaviour of the eMotion-Spheres can be found here: <http://youtu.be/5iqP1oPZ3Qw>

Stereo vision with consumer grade high resolution cameras for a micro air vehicle

Joonas Melin*, Mikko Lauri, and Risto Ritala
Tampere University of Technology, Tampere, Finland

ABSTRACT

We construct a stereo vision system mounted on a micro air vehicle applying two high resolution consumer grade cameras. The system hardware and mechanical configuration are presented. A novel image processing algorithm specifically suited for high resolution stereo images is described, and its properties are discussed. Empirical data from outdoors flight experiments is presented, showing both successful and unsuccessful attempts at extracting depth information from the stereo images. The results highlight several important design aspects to consider when building a stereo vision system for use in micro air vehicles.

1 INTRODUCTION

Depth information is essential for representing real world structures in three dimensions. Depth information represented as a set of 3D points in a known frame of reference is called a point cloud. Point clouds or other 3D representations are essential for autonomous robot operations, since avoiding obstacles, planning routes or manipulating objects in the environment is impossible without 3D information about the surroundings.

Depth data can be obtained, for example, with laser range finders or time of flight (TOF) cameras. However, these methods may not be suitable for micro air vehicles (MAVs) due to weight and cost limitations.

Stereo vision provides point clouds by first finding the matching features from two images taken from two known but different locations. The difference between the matching features is called disparity. Disparity can be converted to a metric distance when the geometric transformation between the images is known. The features are usually block shaped areas in the images. The most difficult part in stereo vision is finding the matching features in the image pairs. Many of the current methods struggle with featureless surfaces, see e.g. [1]

Low resolution stereo imaging as used in [2] and [3] has the advantage of fast processing and low latency, critical for real time applications. For instance, low-resolution imaging

can be used for attitude estimation and navigation when distance to the features is generally under 5m and surfaces have unique texture. On the other hand, these systems are not ideal for mapping areas with small featured surfaces from more than 5m away since the low resolution cameras cannot distinguish texture at distance. This performance is highly dependent on the construction of the stereo camera and the surface in question: plainly painted wall might not have enough features no matter how close the camera is, and the wall of the building with large features might have enough texture for low resolution stereo matching even from more than 20m away. High-resolution stereo imaging can be applied to obtain reliable depth information from feature poor environments such as sand, grass or asphalt surfaces. Another advantage of high resolution images is that they can also be used for tasks such as texture analysis or marker detection. The downside is increased computation time, which makes high resolution data challenging to use in real time applications. In this work, we apply a lightweight, low-cost, stereo camera system built from high-resolution consumer grade cameras to collect depth data. The stereo vision system is mounted on a MAV. Our goal is to apply the system to support the activities of an unmanned ground vehicle (UGV) by providing information on terrain features and formation in areas that the UGVs own sensors cannot reach. The remainder of the paper is organized as follows: in Section 2, we present the MAV hardware construction including the stereo camera system and give a brief overview of the theory of stereo imaging. In Section 3, the image processing techniques developed in this work specific to the use of high-resolution cameras are presented. Their differences to other state-of-the-art methods are discussed. In Section 4, empirical data collected from MAV test flights with the stereo camera system are presented. The data highlight several important design aspects that have an effect on stereo imaging performance. Finally, Section 5 concludes the paper.

2 METHODS

The MAV applied in this work is shown in Figure 1. It is built from carbon fiber honeycomb plate and carbon fiber tubing. The arms and their connections are from a commercial kit but the rest of the hardware has been cut from carbon fiber plates.

The system consists of subsystems which each handle their specific tasks. The main PC coordinates all the subprocesses. The layout and communication hierarchy is presented

*Email address(es): joonas.melin@tut.fi, mikko.lauri@tut.fi, risto.ritala@tut.fi



Figure 1: The MAV used in this work.

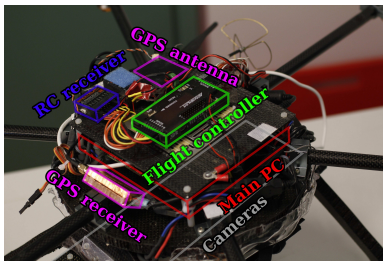


Figure 2: The physical layout of the components of the MAV system.

in Figure 3. The physical placement of the systems can be seen in Figure 2.

We are using the open source Arducopter [4] as our flight controller. An additional inertial measurement unit (IMU) is used for the camera attitude estimation. The IMU unit is built from Arm M4 microcontroller and InvenSense MPU-6050 accelerometer and gyroscope. MAV location is sensed with a global positioning system (GPS) sensor. The GPS sensor consists of a Yuan10 receiver with RTKlib open source RTK-GPS software[5] in the main PC. The positional error of the GPS is usually smaller than 0.5m in motion, but can be as high as 10m when close to ground or buildings. The closer analysis of the positioning system can be found in [6].

The stereo camera is built by mounting two standard consumer cameras on a sandwiched honeycomb composite plate. This mounting ensures that the cameras share a common plane. A similar setup is used e.g. in [7]. Our stereo rig has a baseline of 219 mm, compared with 700 mm in [7]. We use a small baseline because our MAV could not accommodate a larger camera rig. The cameras are running custom firmware that provides software triggering, raw image capturing and basic scripting capabilities. The cameras are triggered by a simple script running in the camera firmware. The camera rig is shown in Figure 4. Cameras are mounted in such a way that we get a maximal baseline with minimal space usage; however this results in a different shutter sweep direction between the cameras. Because we were using the raw images

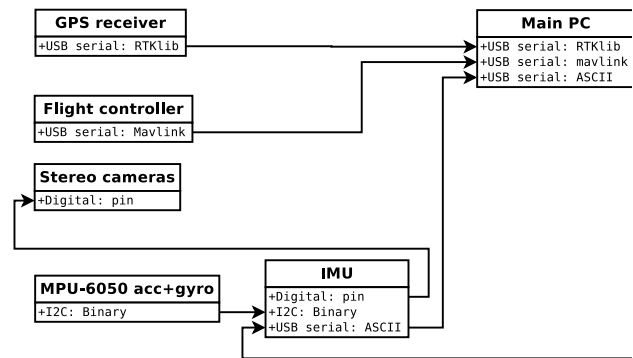


Figure 3: A schematic of the onboard hardware and their connections.

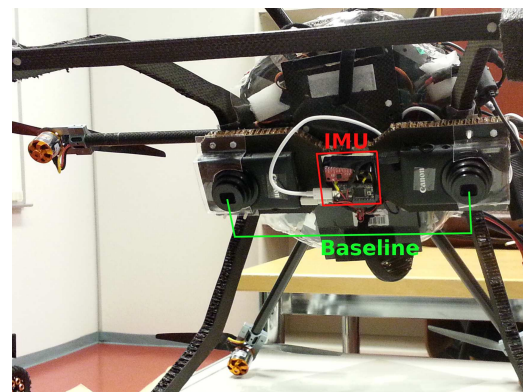


Figure 4: The onboard camera rig.

provided by the custom firmware, we were able to get 10bit images without cropping and other corrections done by the original firmware. However, this also meant that we had to do the factory calibration ourselves, which involved the geometric rectification of the images and removing the dead or stuck pixels from the images. The cameras also had varying triggering delay to the trigger signal. Constant delay was roughly 700ms which was compensated in the camera triggering code, but the variance of this delay proved to be a critical aspect on the experimental data. The triggering difference between the cameras for a sample of ten image pairs is presented in Figure 5.

To perform any kind of mapping, we need to define our coordinate systems and be able to transform data from the sensor plane into the map coordinates. Image processing must be done in a frame relative to the sensor plane and map information is most conveniently represented in a global frame of reference. In the case of the stereo camera, we need at least three coordinate systems: the image, camera and global coordinate systems. The origin of the image coordinate system is at the principal point, which is in the middle of the sensor. Image coordinates have units in pixels. The

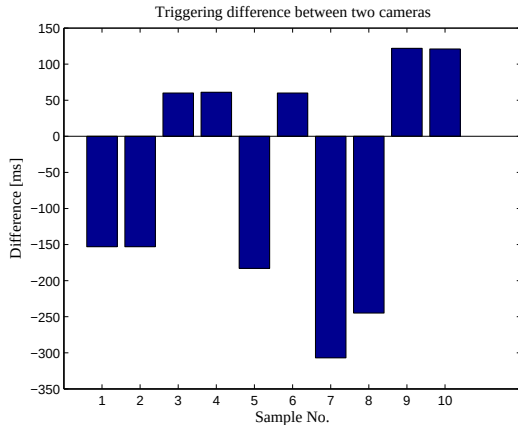


Figure 5: Triggering time differences measured from ten image pairs.

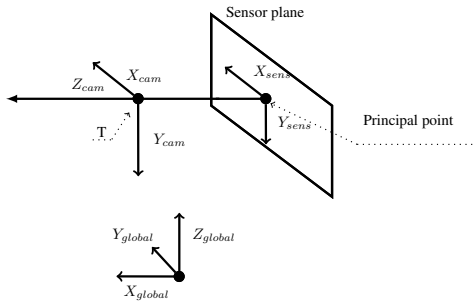


Figure 6: The coordinate systems applied.

camera coordinate system has its origin at the focal point of the camera. The global coordinate system is a fixed earth-centered frame. The coordinate systems are presented in Figure 6, where the coordinate axes with subscript sens, cam and glob represent the image, camera and the global coordinate frames, respectively.

Stereo camera is calibrated to rectify the images so that both of the images appear as if they only have translation in horizontal (X) direction. This means that if a feature is found at coordinate $[x_1, y_1]$ on the left image, it will be found on the right image at a coordinate $[x_1 + d, y_1]$ where the term d is called the disparity. Stereo camera calibration also includes the traditional single camera calibration where the lens distortions are compensated.

There are several methods for finding disparities between images. One of these methods is called block matching. Block matching works by selecting a small rectangular block from the left image and slides it horizontally over the right image, calculating correlation over each position. An alternative version of this typically used in low-resolution systems is the sum of absolute differences, where instead of correlation, difference between the blocks in the images is used. The

point of maximal correlation or the minimal point of summed absolute differences is chosen as the disparity between the images at the origin of the block.

Once the disparity between the images has been calculated at several points, the pixel units in disparities d are converted to metric depth z according to [8, p. 175]

$$z = \frac{d_t f}{d}. \quad (1)$$

This equation is valid in the special case when the cameras are on the same plane. The baseline d_t is the distance between the camera centers. When this information is combined with the focal length f and disparity d we can calculate depth in meters. The focal length f is determined by camera calibration in pixel units.

By this process, we have obtained the metric depth data on the camera coordinate frame. The data is then converted to the global coordinate frame. This requires knowing the rotation and translation of the cameras when the images were taken. Sensor data from GPS, IMU, etc. are used to estimate the rotation and translation parameters.

3 IMAGE PROCESSING

Image processing was first tested with readily available tools like the machine vision toolbox [9]. However, these tools were not always able to handle high resolution data or resulted in low quality image matching. Due to this, we implemented our own stereo matching algorithm specifically suitable for high-resolution images.

We apply a block matching algorithm to find the disparities between the images. Our approach works iteratively, alternating the block size and using the information gained on previous steps to constrain the search areas. The idea behind this method was to utilize all available data with little concern at computation times and concentrate on computational efficiency at a later stage. The algorithm is described in Figure 7. The process begins by loading the images from the disk, followed by the camera calibration procedures which involve the geometric rectification and correcting for faulty pixels. After the calibration steps, we use principal component analysis (PCA) to reduce the RGB color channels to a monochrome image. The disparity between the images is searched with the block matching algorithm. When the disparity is found, the disparity image is converted to point cloud format in global coordinates.

3.1 RGB to monochrome conversion

Stereo vision algorithms are generally implemented on monochrome images as this reduces calculation time. Therefore, our RGB data needs to be converted to monochrome. Traditionally, the RGB image is converted to monochrome by summing the color channels with different gains, roughly equivalent to discarding hue and saturation information from the image. We applied windowed PCA [10] over the whole image in this work. This method finds optimal multipliers

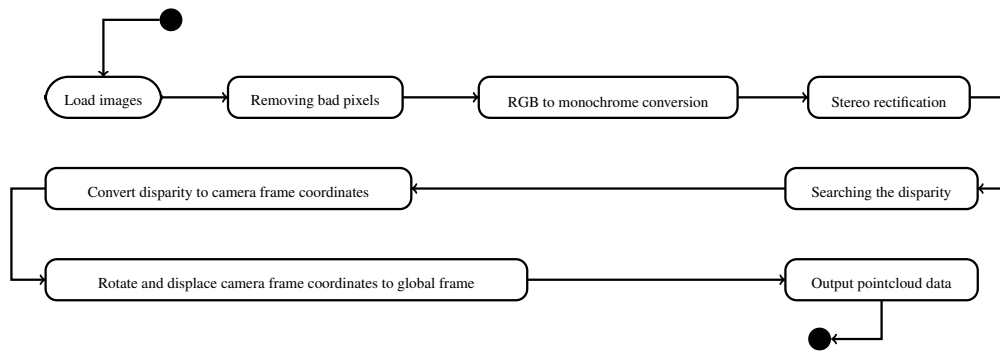


Figure 7: The algorithm for calculating point clouds from stereo images.

for each color channel by computing the PCA transformation to each window so that local contrast is maximized. Comparison of the traditional monochrome conversion and PCA transformation can be seen in Figure 8. The figure shows the worst case scenario for the traditional RGB conversion and has been designed to illustrate a special case when the traditional conversion is not viable. The figure also includes a comparison with a real photograph. Differences are visible in places where colors such as red and green are close to each other. Traditional conversion results in low contrast between the two colors whereas PCA transformation enhances the differences.

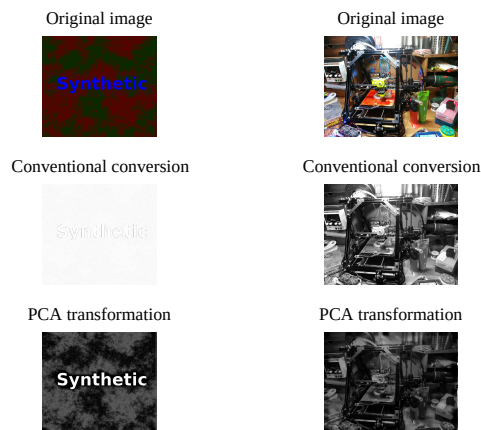


Figure 8: RGB to monochrome conversion for a synthetic (left column) and real image (right column). The source image is shown on the top row. The middle row shows the traditional RGB conversion of the source image and the bottom row the conversion with the windowed PCA transform.

The processing takes 5 seconds for an 8Mpix image with and I7-3990 processor using a window size of 40 pixels. The algorithm was implemented using Matlab's builtin PCA transformation.

3.2 Block matching

We use a block matching algorithm for solving the disparity between the images. A diagram of our block matching algorithm is shown in Figure 9. This algorithm corresponds to the Searching the disparity part of the whole image processing procedure shown in Figure 7. The block matching algorithm works iteratively by first calculating an initial estimate for the image using low resolution and large block sizes. The following iterations will use decreasing block sizes and higher resolution. The decision to replace the results from the previous rounds is done based on cost functions that determine the quality of the match. The algorithm is essentially correlation based block matching with heuristic decision making between different block sizes. The algorithm begins by setting initial values for disparity and the uncertainty of each pixel. In the next phase, the algorithm chooses the block size and resolution to be used in calculation. The search window in the other image will scale according to the uncertainty for that particular disparity. Then, normalized cross correlation is used to calculate the correlation of the block over the search window. This is followed by calculating the different cost functions or descriptors for the disparity. The rest of the algorithm computes heuristic values based on the cost functions for deciding if the disparity calculated in this iteration is better than the one calculated last time. The decision logic is illustrated in the lower part of the Figure 7. For example, in the case where two out of four of the cost functions indicate that the new disparity value would be better than the previously calculated disparity, the old value will be updated with a new disparity value if the window size used was smaller for the new window. The old estimate will also be updated if three or more of the four cost functions indicate that the new estimate is better. The whole algorithm will iterate as long as the percentage of points updated is over a stopping threshold.

The cost functions used in this work fall into two categories. The first category contains pixel based functions, and the other category contains functions that need to be converted to pixels from an unitless quantity, such as correlation.

Uncertainty in pixel units can be converted to a metric uncertainty through the same equations that are used for converting the disparity.

Pixel based cost functions penalize the match if it has too many peaks that are caused e.g. by repetitive texture. Results from larger block size will be preferred if they do not have the multi peak correlation. Standard deviation with the neighboring disparity values is used to penalize noisy pixel values. In other words, smooth estimates are preferred. One cost function is the pixel size. Because we drop the image resolution for the larger block sizes, the pixels will be larger. This results in different pixel sizes between block sizes and the cost function encodes a preference of smaller block sizes over larger blocks. The non pixel based cost functions were the correlation of the found peak, and the signal to noise ratio of the peak. They were converted to the pixel scale by constant multipliers, which were experimentally determined. This ratio is calculated by dividing the correlation peak height with the standard deviation. This ensures that high correlations that clearly stand out from the surroundings are preferred over a high correlation only, which might result from simply flat colored surfaces.

The algorithm we developed resembles the semi-global block matching (SGM) technique used by Hirschmuller et al. [11]. The key difference between our algorithms is in the cost functions used to quantify the match quality. Hirschmuller et al. use only local standard deviation to penalize bad matches whereas our algorithm employs multiple cost functions to describe the match quality. The iterative structure where results from different block sizes are used also differs. We are using results from multiple levels to ensure that even large surfaces without many distinctive features will get matched correctly. With a small window size, these large surfaces will not be properly matched as the small window does not contain enough texture for finding the match. Using large windows mitigates this problem although it creates new problems with small details being lost on the edges. This is why decision heuristics are implemented to favor good matches from smaller block sizes if it is found on the area of the larger block.

3.3 Performance

The algorithm was implemented so that it could be distributed to multiple computers acting as a calculation cluster. The software was running on Matlab and the calculation was distributed through serialized structures sent through ZeroMQ [12], an open source library providing a transport layer for distributed applications.

Average calculation times in a cluster with two Intel i7 3770 processors and Nvidia GTX680 graphics cards was 15-20 minutes for complete processing of one image pair. The current code has not been performance optimized, but it will scale well with the addition of more hardware to calculation. As the algorithm runs iteratively, initial results are available

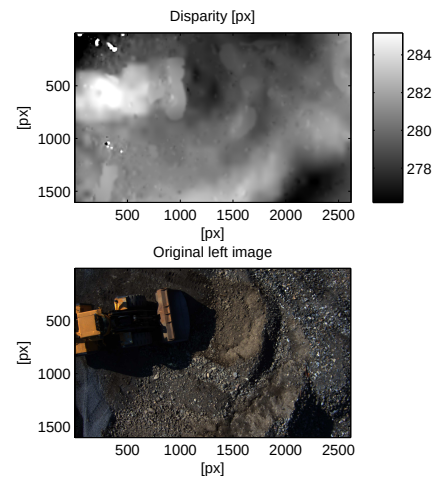


Figure 10: Original image and disparity extracted from the stereo pair in good conditions.

in under one minute with the current setup. For the rest of the time, the estimate is refined to produce a more accurate end result.

4 EXPERIMENTAL RESULTS

In this section, we present some of the best and worst results obtained with our algorithm. Comparing the end results highlights important design aspects that must be taken into account when designing a stereo vision system for use in a MAV. We also present a comparison of our algorithm to the SGM technique of [13]. Figure 10 presents the performance of the disparity calculation in ideal conditions when there is sufficient texture and the triggering of the cameras does not have large time difference. This results in a smooth disparity estimate with only a small amount of missing data due to bad matching.

Figure 11 presents the point cloud extracted from the disparity image seen in Figure 10. Points on the lower left edge are missing due to estimate being too inaccurate. Small errors are also present on the upper right corner where the spike in height is not present in the real terrain. Axes are scaled to 10 m. There is no visible warping on the point cloud caused by the inaccurate triggering of the cameras.

Figure 12 present the system's performance in poor conditions. The matching itself is good but there is a clear trend visible towards the lower left corner. This is a good example of a case where the assumed camera calibration is invalid because cameras were triggered at different times and the position and attitude of the cameras does not correspond to the calibrated positions. This often results in a valid looking disparity image, but the point cloud reveals heavy distortions. Compensating was attempted by forcing the mean disparity to correspond to the height of the MAV but this correction only applies to the incorrect baseline and does not fix the possible

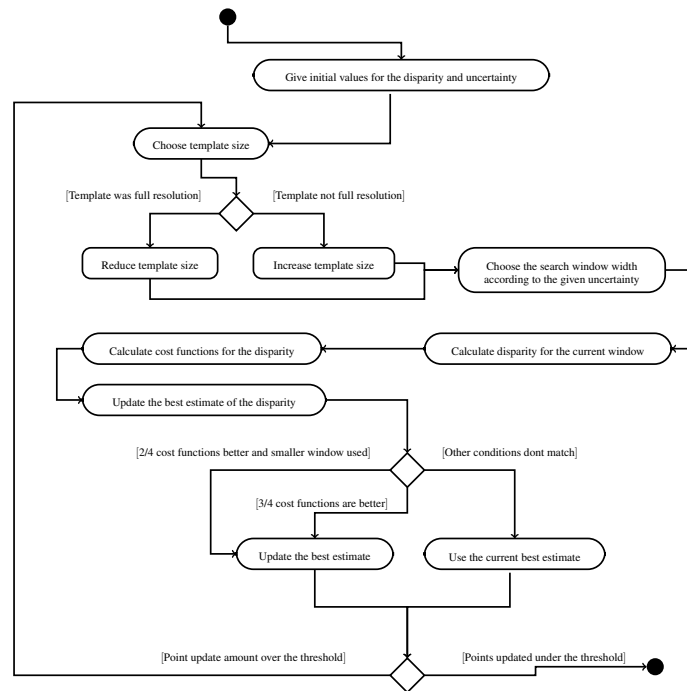


Figure 9: An activity diagram for the disparity search algorithm. Each of the operations is performed for the whole image.

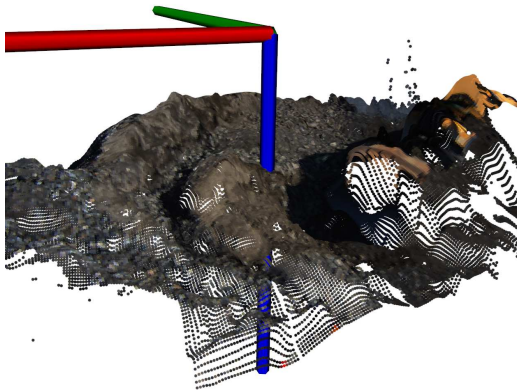


Figure 11: Point cloud extracted from disparity image.

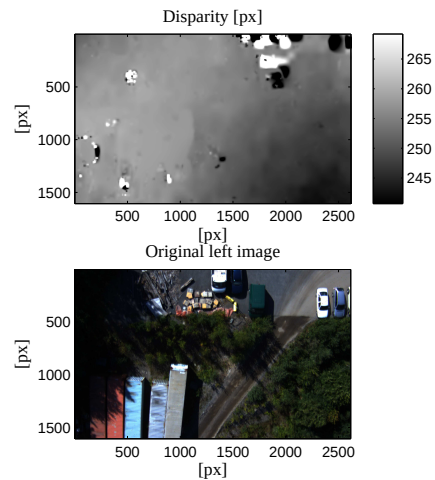


Figure 12: Original image and the disparity extracted from a stereo image pair in poor conditions.

camera rotation between the images. Knowledge of the exact camera triggering times would be required to correct for these errors.

Figure 13 shows the point cloud extracted from the disparity image in Figure 12. The effect of assuming an invalid calibration is clearly visible in this figure, as can be seen from the distorted surface. In reality, the ground plane seen in the figure should be flat.

We tested our algorithms performance against an implementation of SGM [14]. According to the software documentation, the implementation is based on [13]. The comparison is presented in Figure 14. We used the image pair that was earlier presented in Figure 10. We used a block size of 17px, with search window size of 64x304px with other values left at their default values. It can be seen that our method man-



Figure 13: A point cloud extracted from a disparity image when an invalid camera calibration is applied.

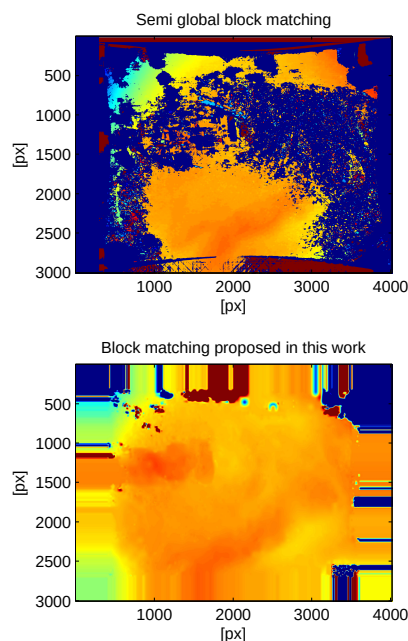


Figure 14: Comparing Matlab implementation of semi global block matching to our algorithm.

ages to find a more complete disparity estimate. However, it is worth noting that SGM took 12seconds to calculate the results on a single PC as our method took roughly 15 minutes running on two processors and one GPU. The result indicates that there may be cases where our algorithm outperforms SGM. However, a fair comparison requires testing with more input images and is a topic for future work.

5 CONCLUSION

We applied high-resolution, low-cost consumer grade cameras to build a lightweight stereo camera system mounted on a MAV. Specialized algorithms were developed to extract depth information from the high-resolution images. We were able to extract high quality depth data from low textured surfaces like dirt roads, sand piles and grass. High resolution stereo vision was useful in identifying features and making the system compact without making the field of view too narrow.

In flight altitudes exceeding 10m, the quality of the point clouds decreased. This was due to the features being too small. This can be seen as increasingly noisy depth estimates for low featured surfaces. Another drawback was caused by the use of the consumer grade cameras which did not trigger accurately at the same time. This resulted in the stereo calibration assumed in the image processing phase being inaccurate. Inaccuracy in camera calibration was often not critical when the altitude was low as the disparities were small, but when altitude was more than 10m, the results degraded heavily. We identified several important factors that should be taken into account when designing a stereo camera system for use in a MAV. The single most important factor is to trigger the cameras at precisely the same time. Even differences as small as 100ms can cause significant errors when the cameras are moving at typical MAV speeds. Using multiple block sizes proved to be useful while matching images with low texture. High computation times were expected as the goal of this work was not to make a real time stereo vision system, but to explore the possibilities of the high resolution stereo vision. This work is a summary of one of the authors Masters thesis [15]. The algorithm development and data gathering was done during year 2013. In future work, we will concentrate on using machine vision cameras with accurate hardware-level triggering. We are also planning to test how low textured surfaces can be matched using low resolution cameras with a narrow field of view. In addition, computation is faster for lower resolution images. Smaller cameras also enable us to use a longer baseline to compensate for the smaller resolution.

ACKNOWLEDGEMENTS

The authors would like to thank Mr. Heimo Ihalainen for discussions and valuable insights related to this work.

REFERENCES

- [1] H. Sunyoto, W. Van der Mark, and D.M. Gavrilu. A comparative study of fast dense stereo vision algorithms. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 319–324, June 2004.
- [2] K. Schmid and H. Hirschmuller. Stereo vision and imu based real-time ego-motion and depth image computation on a handheld device. In *Robotics and Automation*

- (ICRA), *2013 IEEE International Conference on*, pages 4671–4678, May 2013.
- [3] K. Schmid, T. Tomic, F. Ruess, H. Hirschmuller, and M. Suppa. Stereo vision based indoor/outdoor navigation for flying robots. In *Intelligent Robots and Systems (IROS)*, *2013 IEEE/RSJ International Conference on*, pages 3955–3962, Nov 2013.
 - [4] Arducopter project. Arduino-based autopilot for mulitrotor craft. Available online: <http://copter.ardupilot.com/>, 2014.
 - [5] T. Takasu. RTKLIB: An open source program package for gnss positioning. Available online: <http://www.rtklib.com/>, 2014.
 - [6] R. Hellevaara. RTK-GPS on a UAV Platform: Determining Coordinates from an Image (in Finnish) . Master's thesis, Tampere University of Technology, Finland, 2013.
 - [7] M. Warren and B. Upcroft. High altitude stereo visual odometry. In *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013.
 - [8] O. Faugeras. *Three-dimensional computer vision : a geometric viewpoint*. MIT Press, Cambridge, Mass, 1993.
 - [9] Peter I. Corke. *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011.
 - [10] C. Bishop. *Pattern recognition and machine learning*. Springer, New York, 2006.
 - [11] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):328–341, 2008.
 - [12] iMatix Corporation. ZeroMQ intelligent transport layer for distributed apps. Available online: <http://zeromq.org/>, 2014.
 - [13] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE, 2005.
 - [14] MATLAB. *version 8.3 (R2014a)*. The MathWorks Inc., Natick, Massachusetts, 2014.
 - [15] J. Melin. Surface reconstruction using high resolution stereo vision in a micro air vehicle. Master's thesis, Tampere University of Technology, Finland, 2014.

Fullscale Windtunnel Investigation of Actuator Effectiveness during Stationary Flight within the Entire Flight Envelope of a Tiltwing MAV

M. Schütt*, P. Hartmann† and D. Moormann

Institute of Flight System Dynamics, RWTH Aachen University, Germany

ABSTRACT

The tiltwing aircraft developed by team MAVerix of RWTH Aachen University is intended to operate in a continuous range of horizontal velocities from hover to fixed wing flight. The design of a flight controller enabling stationary flight states within the entire transition flight envelope demands detailed knowledge of the aircraft's flight mechanics. Data to implement a feed forward controller for longitudinal actuator settings (tilt angle, thrust and elevator/tail thrust) to satisfy stationary operating points in all horizontal velocities of the transition are gained by a wind tunnel investigation in the presented approach. Furthermore, to perform attitude controller design, the effectiveness of all actuators is recorded to decouple the attitude controller from altering magnitudes and directions of the actuators effects during transition.

NOMENCLATURE

L, D	Lift and drag of the entire aircraft
T	Thrust specified by index
U_∞	Freestream velocity
X, Z	forces in x - and z -direction
L	Rolling moment
M	Pitching moment
N	Yawing moment
q	Dynamic pressure
b	Halfspan of the wing
S_{Wi}	Wing surface
x, y, z	Axis in coordinate system specified by index
α	Angle of attack
η	Elevator deflection
η_p	Tail rotor pitch deflection
δ	Differential thrust
ξ	Aileron deflection
σ	Tilt angle

*Email address: marten.schuettt@rwth-aachen.de

†Email address: hartmann@fsd.rwth-aachen.de

Indexes

w	Weighing scale coordinate system
a	Aerodynamic coordinate system
f	Body fixed coordinate system
g	Earth coordinate system
Main	Main propulsion system
Tail	Tail rotor

1 INTRODUCTION

Team MAVerix is a student team from RWTH Aachen University participating in the outdoor-challenge of the International Micro Air Vehicle conference and competition (IMAV) since 2011. With respect to various mission challenges, the contestant Micro Air Vehicle (MAV) was designed as a tiltwing aircraft. In the course of past competitions, it was not possible to make use of the tiltwings full flight performance potential, due to deficiencies in attitude control. The MAV was not able to perform transition from hover to purely aerodynamic flight reliably. Control deviation due to unknown and therefore unconsidered coupling in actuator effectiveness was identified as one of the main reasons for this issue. Thus, an attitude controller including explicit decou-

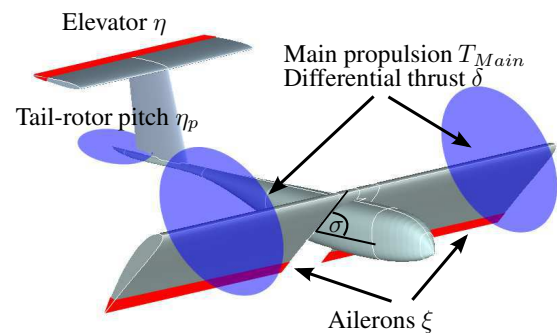


Figure 1: MAVerix with actuators

pling of actuator effectiveness was designed. To gain the required understanding of actuator effectiveness for all stationary flight states, a full-scale windtunnel investigation was carried out.

This contribution presents the corresponding windtunnel

setup as well as selected results on actuator effectiveness. As a prerequisite for this investigation, actuator deflections for trimmed stationary flight within the entire flight envelope were measured and will also be presented.

2 TILTWING AIRCRAFT

The tiltwing configuration forms a combination of two different concepts of aircraft that allows stationary flight in a wide range of velocities. The advantages of a rotary-wing and a fixed-wing configuration are joined to perform energy efficient high speed flight in combination with vertical take-off and landing (VTOL) capabilities. The propulsion system is fixed to the wing so that the ailerons stay in slipstream of the propeller. For VTOL and hovering the wing is rotated and the propulsion system produces the lift. To perform high speed flight efficiently, lift is generated by the wing in fixed-wing mode.

2.1 Specification of examined MAV

The MAV with a wingspan of 0.96 m and a wing area of 0.24 m has a mass of ≈ 1.7 kg. The main propulsion system consists of two counter rotating brushless electric motors. The MAV is powered by a 4-cell lithium polymer battery. The ailerons extend over the complete size of the wingspan to use most of the slipstream. The MAV is equipped with a horizontal tail rotor to balance the pitching moment. The thrust of the tail rotor is varied by changing the pitch angle of the rotor blades while the rotational speed stays constantly at 7300 RPM. For reasons of simplification the rudder was omitted. In Figure 1 the actuators of the MAV are displayed.

2.2 Flight mechanics

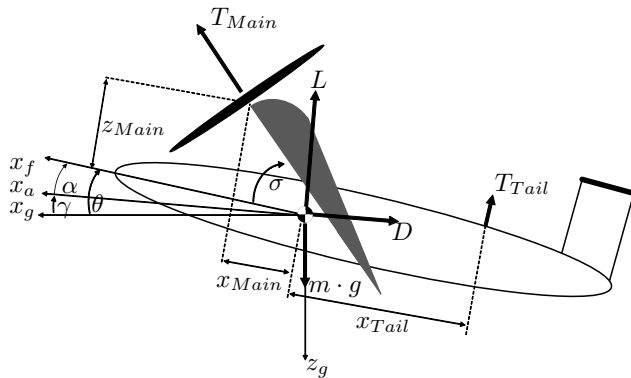


Figure 2: Forces and moments in longitudinal motion

Regarding flight velocity, the entire flight envelope of the tiltwing can be divided into three regions. While the pure aerodynamic flight and hover configurations form the border

areas of the flight envelope, a third configuration, the transition, covers all flight states in-between.

The fixed-wing configuration is characterised by a fixed horizontal wing (tilt angle $\sigma=0$, compare figure 2) and well-known flight mechanics. Ailerons induce a rolling-, rudder (in our special case a differential thrust, compare figure 1) induce a yawing- and elevators induce pitching moments.

In hover configuration the wing is fixed in a vertical position. Flight mechanics can be compared to some extent to multicopter flight mechanics. Differences can be found for example in steering of yawing moment by deflection of the ailerons in the slipstream of the main propulsion system.

The transition is the phase in-between hover and aerodynamic flight. Depending on the tilt angle, differential thrust ΔT_{Main} of the main propulsion system as well as the ailerons ξ cause a combination of a rolling and a yawing moment. Differential thrust produces a rolling moment in hover and a yawing moment in aerodynamic flight. Analogous, as the ailerons ξ stay in the slipstream, they produce a yawing moment in hover and a rolling moment in aerodynamic flight. Not only is the direction of the effect dependent on the tilt angle but also is the effects magnitude due to different dynamic pressure q associated with airflow velocities. A simplified description of yawing and rolling moments can be found in (1) and (2). The general sign convention, that an actuator deflection is defined as positive when producing a negative moment [1], cannot be satisfied with a tiltwing. The ailerons cause a positive yawing moment in hover.

$$L = -2 \cdot q(U_{\infty}, T_{Main}) \cdot \Delta C_A(\xi) \cdot S_{Wi} \cdot \frac{b}{2} \cdot \cos(\sigma) - 2 \cdot \Delta T_{Main} \cdot y_{Main} \cdot \sin(\sigma) \quad (1)$$

$$N = -2 \cdot \Delta T_{Main} \cdot y_{Main} \cdot \cos(\sigma) + 2 \cdot q(U_{\infty}, T_{Main}) \cdot \Delta C_A(\xi) \cdot S_{Wi} \cdot \frac{b}{2} \cdot \sin(\sigma) \quad (2)$$

In summary, with regard to the feedback control problem, it can be stated that the lateral steering during the transition flight state is heavily coupled.

Also the elevators effect depends on airflow velocity. For small forward velocities the elevator does not produce enough effect to balance the pitching moment, therefore an additional force is required. Furthermore during transition nonlinear effects like stall and couplings can cause instabilities.

Because of these effects the transition is often performed as a single continuous process by tiltwing, tiltrotor or tail-sitter aircraft. Transition trajectories are developed to get from hover to aerodynamic flight in a time optimized way [2]. However, the advantage of a tiltwing configuration - a continuous velocity range - can only be used by satisfying stationary flight states in all horizontal velocities of the flight envelope [3].

For a stationary flight state during the transition the forces and moments of the longitudinal motion have to be in an equilibrium $X, Z, M \stackrel{!}{=} 0$. Hereafter this state is called an operating point. The main thrust of both motors is summed up to $T_{\text{Main}} = T_{\text{left}} + T_{\text{right}}$. The three equations in (3) depend on the tilt angle σ and were simplified for small angle of attack α and pitch angle θ . The acting forces and moments are considered in the body-fixed coordinate system as displayed in figure 2. With changing horizontal velocity and tilt angle the force balancing the weight is altered from thrust to aerodynamic lift.

$$\begin{aligned}\Sigma X_f &= T_{\text{Main}} \cdot \cos(\sigma) - D(U_\infty, \sigma, T_{\text{Main}}) \stackrel{!}{=} 0 \\ \Sigma Z_f &= -T_{\text{Main}} \cdot \sin(\sigma) - T_{\text{Tail}} + m \cdot g \\ &\quad - L(U_\infty, \sigma, T_{\text{Main}}) \stackrel{!}{=} 0 \\ \Sigma M_f &= M_{\text{Aero}} + T_{\text{Main}} \cdot (\sin(\sigma) \cdot x_{\text{Main}} \\ &\quad - \cos(\sigma) \cdot z_{\text{Main}}) - T_{\text{Tail}} \cdot x_{\text{Tail}} \stackrel{!}{=} 0\end{aligned}\quad (3)$$

2.3 Feedback Control

Historically the transition has always been hard to control due to the instabilities and non-linearities[4]. Thanks to more complex control algorithms the transition is controllable today.

For the MAV tilting a unified linear controller was designed. The controller negates the distinction between the three regions hover, transition and aerodynamic flight. The transition is treated as the general case, aerodynamic flight and hover are only border cases of transition. To be able to satisfy stationary, trimmed flight in all velocities within the transition, stationary operating points have to be known. A mapping and feed forward control of the deflection of all actuators for stationary longitudinal motion for different horizontal velocities is planned.

The AVIGLE tilting [5] uses a controller adjusting the tilt angle depending on the desired horizontal velocity while the thrust is controlled by the altitude control. HARVee is a Tiltwing [6] performing controlled transition flight based on a linearized model, also using automatic tilt control only. A feed forward control of the thrust and the pitching moment depending on the horizontal speed for tilting aircraft cannot be found in the literature.

For a unified attitude controller design, an explicit compensation of different actuator couplings and actuators effectiveness is required. The decoupling by inverting the actuator effects is ideally done before control design [7]. Therefore the variation of the actuator effectiveness in different operating points has to be analysed. With the inversion of actuator effects the attitude controller is approximately not affected by couplings and variation of the actuators and can be designed based on several independent single input single output (SISO) systems.

3 WIND TUNNEL MEASUREMENTS

The variances and couplings during the transition are investigated in wind tunnel measurements. The required settings for the actuators of the longitudinal motion for reaching a stationary operating point for different horizontal velocities are investigated. At all operating points all actuators are varied to record their effects direction and magnitude.

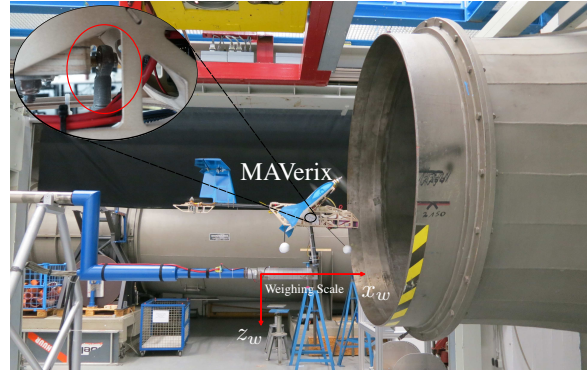


Figure 3: The measurement setup

The Institute of Flight System Dynamics operates a closed circuit wind tunnel with an open measurement section with a diameter of 1.5 m. The freestream velocity can be adjusted from 0 – 70 m/s and was limited to 20 m/s during investigation. Due to the small size of the MAV a full scale airworthy measurement with simulated flight in the wind tunnel can be performed. The MAV is fixed on a weighing scale measuring resulting forces and moments. The setup is presented in figure 3, the fixed attachment can be seen as well as the ball joint as described in section 4.3. The pitch angle is set to zero as the airplane shall stay in horizontal position for all operating points of the transition. The freestream velocity correlates to the forward velocity of the MAV.

3.1 Investigation of Operating Point Actuator Deflections

Calibration of the weighing scale and reset to zero was done without the MAV installed on it. As a consequence, the weighing scale measured all forces including force of gravity, thus fulfilment of equations in (3) can easily be checked. By variation of the tilt angle σ , the thrust setting T_{Main} , deflections of the elevator η and variation of the vertical thrust T_{Tail} stationary operating points for all horizontal velocities of interest are found. The variations of the actuators are manually commanded by a remote control.

Figure 4 illustrates the procedure of reaching an operating point at an exemplary freestream velocity of 13 m/s by eliminating residual forces. The operating point found is tagged with a circle. Increase of the tilt angle causes a decrease of the force in z-direction and increasing thrust results in an increasing force in x-direction. With changing tilt angle the impact of changing thrust and tilt angle alters. This procedure is

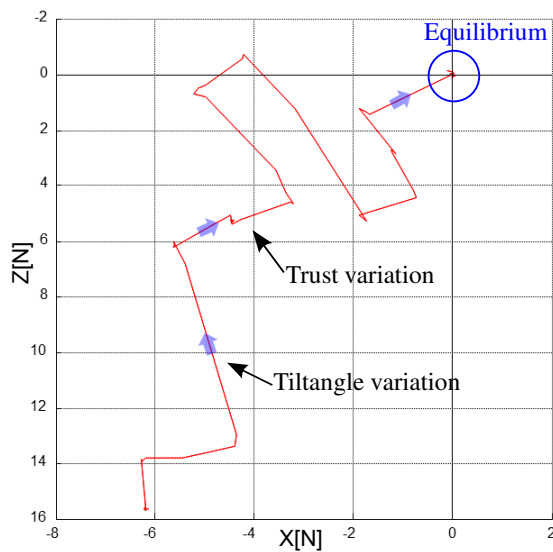


Figure 4: Process of balancing residual forces dependent on variation of tilt angle and thrust during exemplary investigation on one operating point

repeated for all velocity sample points within the flight envelope. Associated results are given in section 4.1

3.2 Investigation of Actuator Effectiveness

After finding an operating point for a particular velocity, the effectiveness in magnitude and direction of all actuators is measured. Dynamics of the actuators are not taken into account as only stationary measurements are recorded. As the attitude controller controls the tilting by generating rolling, pitching and yawing moments, the investigation focusses on the related actuators differential thrust, aileron deflection, elevator deflection and thrust variation of the tail rotor. These actuators are varied in two directions and two different amplitudes. For evaluation, the actuator effectiveness is normalized to unitary actuator deflection afterwards. The effectiveness of all actuators of interest is symmetric and assumed to be linear. Associated results are given in section 4.2

4 RESULTS

At first the actuator deflections for longitudinal motion to obtain stationary operating points at different horizontal velocities are presented. The alteration of actuator effectiveness is discussed subsequently. All measurement plots are equipped with a compensating graph. The compensation function is given by a polynomial of different orders as a trade-off between smoothing and good reproduction of the measurement.

4.1 Trim Point Actuator Deflections

Figure 5 shows the operating point tilt angle as function of freestream velocity. The tilt angle decreases with

increasing horizontal velocity. The entire range of possible tilt angles is not used, as a tilt angle of 90° produces a force in negative x-direction due to the propeller induced lift of the wing. On the one hand the tilt angle represents the direction of thrust, on the other hand the direction of the lift force. The magnitude of the lift increases with the increasing horizontal velocity. The trajectory is very similar to a simulated transition scheduling trajectory for the tilting HARVee [2].

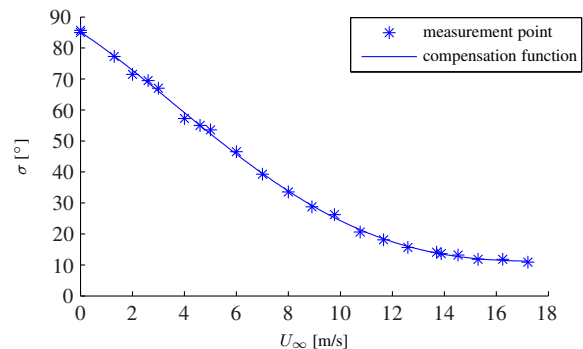


Figure 5: Operating point tilt angle σ for different horizontal velocities U_∞

In figure 6 the operating point thrust is given in terms of rotational speed of the main propellers. The input signal of the speed controllers of the main propulsion system was matched to a corresponding rotational speed during ground tests. The resulting deviations to reality because of incoming airflow are tolerated as this is only for overview. The operating point thrust provides the lifting force in hover and decreases while the lift is more and more produced by the wing with increasing horizontal velocity. A minimum of thrust is required at roughly 12 m/s while more thrust is necessary with increasing horizontal velocities due to growing drag.

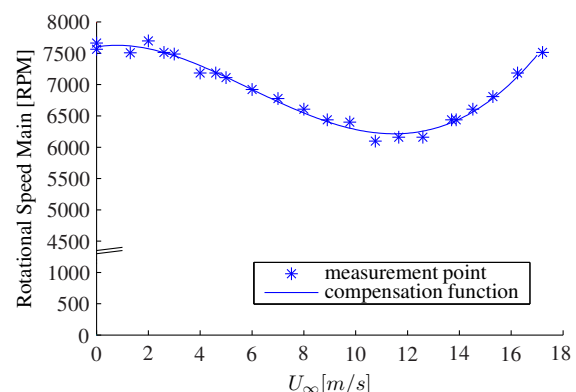


Figure 6: Operating point thrust of the main propulsion system for different horizontal velocities U_∞

Regarding elevator and tail rotor pitch deflections, the measurements in figure 7 show more variations than before. The variations can be explained by small possible movements around the pitching axis despite fixed mounting on the scale. For low horizontal velocities the pitching moment is produced by the tail rotor due to low effectiveness of the elevator. For velocities from 7 to 10 m/s elevator and tail rotor are used in combination. At a horizontal velocity of 10 m/s the effect of the elevator is big enough to operate the aircraft solely with the elevator. A provision in possible deflection for pitch control is available. Beyond this velocity the tail rotor is not necessary any more and is turned off.

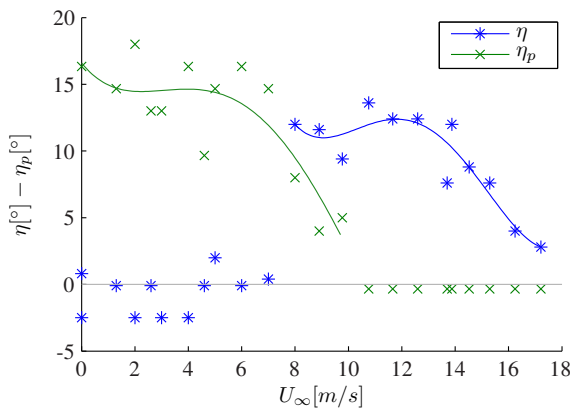


Figure 7: Operating point deflection of the elevator and tail rotor for different horizontal velocities U_∞

For constant power supply conditions the aircraft is powered by a mains power adapter during the wind tunnel measurements. This way the power consumption at different operating points can be analysed and is shown in figure 8. The main propulsion system represents the main power consumer. The shutdown of the tail rotor causes a drop in the trend of the power consumption. The most efficient and longest endurance flight can be accomplished by flying at a horizontal velocity of 12 m/s.

4.2 Actuator Effectiveness

In figure 9 effects of a positive, normalized differential thrust actuation on rolling and yawing moments for different operating points and their corresponding tilt angles are presented. Looking at the yawing and rolling moment as a vector, the vector rotates related with the tilt angle of the wing. This effect approves the approximation in (1) and (2), which are plotted in dashed lines. The thrust ΔT_{Main} is scaled to the un-interfered hover case of $\approx 10 \text{ Nm}$. Deviations from this theoretical approach might be caused by different horizontal velocities that change the inflow of the propellers and therefore the generated force variation. The main impact of this presumed effect can be seen in the yawing moment at fixed

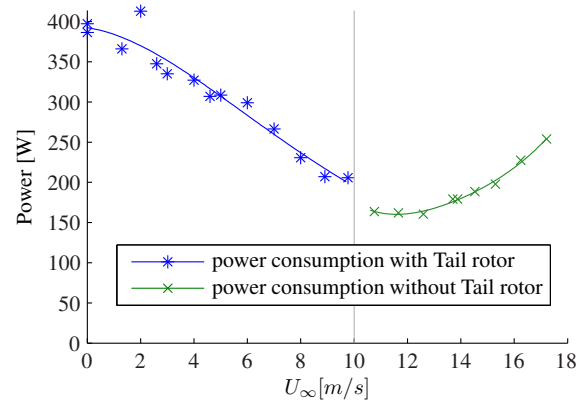


Figure 8: Power consumption at different operating points distinguishing between stopped and running tail rotor

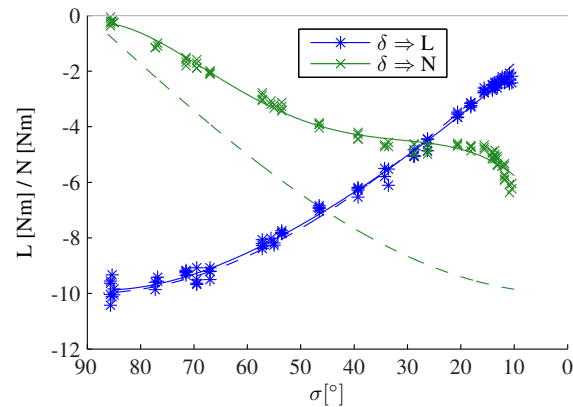


Figure 9: Effects of differential thrust in rolling and yawing moments at different tilt angles

wing flight, which is in fact reduced to $\approx 6 \text{ Nm}$.

In figure 10 the effects of a positive, normalized aileron deflection on rolling and yawing moments for different operating points and their corresponding tilt angles are presented. Similar to the differential thrusts effect, the direction of the aileron deflection effect changes because of the wings rotation. The magnitude of the effect is - as predicted in (1) and (2) - dependent on the dynamic pressure q which is influenced by the freestream velocity and the slipstream of the propulsion system. The approximations of these equations (plotted in dashed lines) are scaled to the hover case again. This leads to assumptions of the dynamic pressure of the propulsion system and the additional quasi-constant parameters. The dynamic pressure of the freestream velocity is matched to the current tilt angle according to figure 5. The neglected variation of the parameters over the change the inflow at different tilt angles and freestream velocities is one reason that leads to the deviation from the experimental results.

In figure 11 the effects of the elevator deflection and the

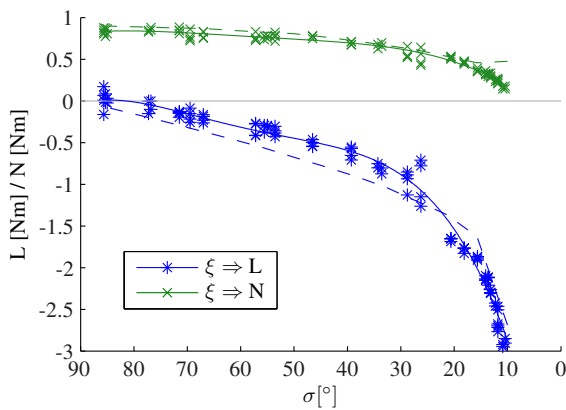


Figure 10: Effects of aileron deflection in rolling and yawing moments at different tilt angles

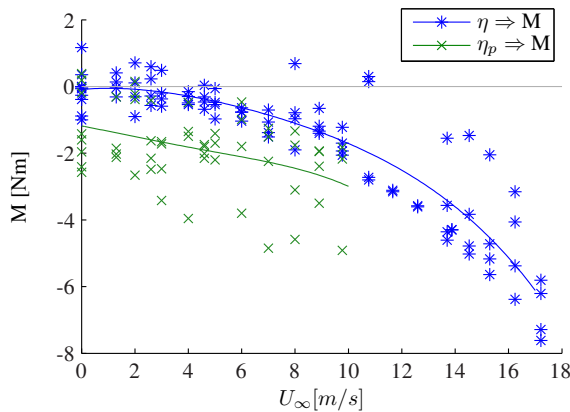


Figure 11: Effects of elevator deflection and tail rotor pitch deflection in pitching moment at different horizontal velocities

tail rotor pitch deflection on pitching moment are presented at different horizontal velocities. Please note, due to the fact that the tilt angle is not the main influence here, the abscissa indicates velocity, not tilt angle. The effect of the elevator is directly depending on the inflow velocity. Influences on the tail rotor effectiveness includes but is not limited to inflow velocity, the tilt angle and thrust of the main propulsion system. In total the influences cause an increasing effect with increasing horizontal velocity.

4.3 Relevance of Results for Controller Design

The wind tunnel investigation reveals the possibility to associate deflections for all actuators controlling the longitudinal motion to horizontal velocities. This can be used for a feed forward controller that steers all mentioned actuators in a way to obtain stationary flight for a given velocity.

The draft version of a feed forward control design for the MAV is realized in MATLAB/SIMULINK using *lookup-tables*

to match the longitudinal actuator settings to a commanded horizontal velocity. Therefore the compensating functions approximating the wind tunnel data are discretized in steps of 0.5 m/s, in-between the sampling points the controller interpolates linearly.

A second wind tunnel measurement to validate the draft version of the feed forward controller was performed. Associated results can be seen in figure 12. During this measurement the aircraft was connected to the weighing scale by a ball joint, allowing rotational movements around the center of gravity of the aircraft. A detailed view of the ball joint is presented in figure 3. With different freestream velocities, while commanding a matching horizontal velocity, the remaining forces in x- and z-direction of the weighing scale were recorded. For all different horizontal velocities of the transition from 0 m/s to 10.1 m/s stationary flight states could successfully be obtained as the remaining forces only vary in a small range around zero.

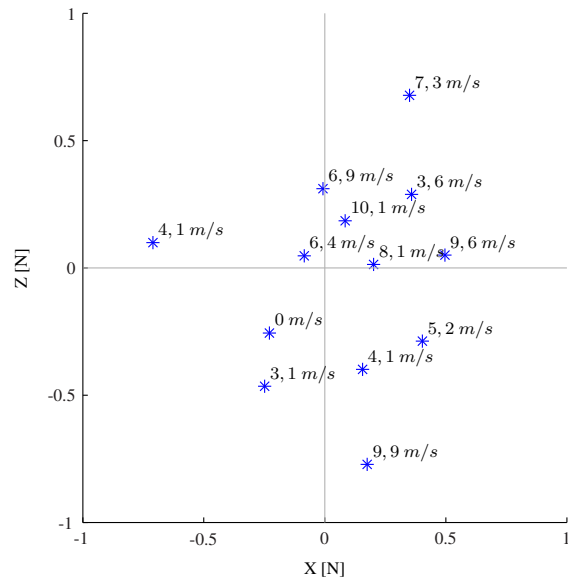


Figure 12: Remaining forces at the operating points at different horizontal velocities showing approximately zero mean residual distribution

Regarding attitude control the measured actuator effectiveness will help to decouple the steering of the tilting MAV. The altering of the actuators effectiveness is continuous over increasing tilt angles respectively increasing flight velocities. Additionally moments generated by ailerons, differential thrust and elevator are linearly independent at all operating points. This facts motivate an attitude controller including inversion of actuator effectiveness aiming at homogenisation of flight characteristics.

5 CONCLUSION

In this paper a wind tunnel investigation to gain insight into the flight mechanics of a tiltwing-aircraft during transition has been presented. Based on this investigation a draft version of a feed forward controller has been implemented. The draft version of the controller allows stationary trimmed flight within the entire flight envelope. The feed forward controller concept has successfully been proven in a second wind tunnel measurement for velocities from 0 m/s to 10 m/s. The aircraft is able to balance all forces and moments at any commanded horizontal velocity in this range. For pure aerodynamic flight with horizontal velocities from 10 m/s to 20 m/s the concept has to be proven in free flight experiments.

For future work the flight envelope can be extended towards small negative velocities during hover flight by making use of the entire tilt angle range up to 90° .

To achieve identical flight characteristics within the complete transition, actuator effects have to be independent of a particular operating point. The alterations in direction and magnitude of the actuator effects dependent on the operating point have been recorded and analyzed. An inversion of these effects may provide an opportunity for designing an attitude controller consisting of several decoupled SISO controllers.

REFERENCES

- [1] R. Brockhaus, W. Alles, and R. Luckner. *Flugregelung*. Springer-Verlag, 2011.
- [2] Jeffrey J. Dickeson, David Miles, Oguzhan Cifdaloz, Valana L. Wells, and Armando A. Rodriguez. Robust LPV H_∞ Gain-Scheduled Hover-to-Cruise Conversion for a Tilt-Wing Rotorcraft in the Presence of CG Variations. *Proceedings of the 2007 American Control Conference*, pages 5266–5271, 2007.
- [3] T. Ostermann, J. Holsten, Y. Dobrev, and D. Moormann. Control Concept of a Tiltwing UAV During Low Speed Manoeuvring. *Congress of the International Council of the Aeronautical Sciences*, 28:1–10, 2012.
- [4] L. P. Thomas III. A flight study of the conversion maneuver of a tiltwing vtol aircraft. Technical note d-153, NASA, 1959.
- [5] J. Holsten, T. Ostermann, Y. Dobrev, and D. Moormann. Model validation of a tiltwing uav in transition phase applying windtunnel investigations. *Congress of the International Council of the Aeronautical Sciences*, 28:1–10, 2012.
- [6] Jeffrey J. Dickeson, Oguzhan Cifdaloz, David Miles, Paul M. Koziol, Valana L. Wells, and Armando A. Rodriguez. Robust H_∞ Gain-Scheduled Conversion for a Tilt-Wing Rotorcraft. *Proceedings of the 45th IEEE Conference on Decision & Control*, pages 5882–5887, 2006.
- [7] Jan Lunze. *Regelungstechnik 2 - Mehrgroßensysteme, Digitale Regelung*. Number Bd. 2. Springer-Verlag, 2010.

Robust marker-tracking system for vision-based autonomous landing of VTOL UAVs

M. Verbandt*, B. Theys† and J. De Schutter‡
 Department of Mechanical Engineering PMA, KU Leuven,
 Leuven, 3000 Belgium

ABSTRACT

This article presents the development of a vision-based VTOL UAV docking system for outdoor applications. A new marker type is constructed, along with a robust detection algorithm based on the Hough transform, which forms the basis of the positioning system. For verification purposes, a binary classifier based on support vector machines is trained. Test data show great robustness against varying lighting, partial occlusion and unsharpness. The practical usefulness is demonstrated by creating a single tracking module, based upon the PX4FLOW camera. Equipping a quadrotor with the new positioning sensor allows it to perform a precision landing.

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are being used by the army for a long time now, but technological advancement is making them more and more available to the larger public. This development opens a door to a variety of new applications, e.g. rescue operations, urgent transportation of goods to remote areas, map-building, etc. One of the major drawbacks is their limited operating time. However, landing on a ground station offers the possibility to automatically replace the empty battery by a charged one, dramatically increasing the operating range and time of the UAV.

The main difficulty in automatic docking is the positioning. Figure 1 sketches the problem. The UAV has to determine its relative position and orientation w.r.t. the landing platform and use this information to carry out a precision docking. [1] is a good example of successful UAV docking in which the main focus was to execute persistent missions involving a hot-battery swap.

Docking mobile robots has been a research-topic for decades, during which many different positioning systems have been developed. Three main categories exist: passive optical systems, active optical systems and ultrasound



Figure 1: Sketch of the docking problem: the UAV has to determine its relative position w.r.t. the landing platform and use this information to land on the target.

systems.

A passive optical positioning system consists of only light-sensitive devices, e.g. digital cameras. Detection of the specific docking target in the image allows estimation of the relative pose. A distinct pattern, called a marker, is usually added to ease the extraction from an image. Figure 2(a) gives an example of a passive optical positioning system. The marker is in this case a checkerboard pattern.

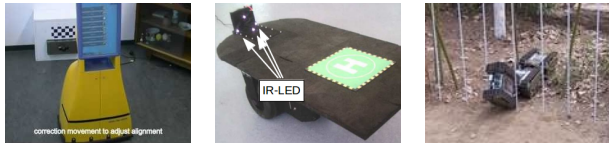
An active optical positioning system however contains additional controllable light sources on the target. Comparing images of the same scene with the sources switched on and off simplifies the extraction of the target. [2] describes a successful application on a UAV (figure 2(b)). Four IR-LEDs are mounted onto a landing platform which are then used as a reference for the UAV's position. A major drawback is that high intensity ambient lighting can overwhelm the source, making it impossible to locate the target.

An ultrasound positioning system, on the other hand, uses ultrasonic waves. Depending on the specific arrangement, multiple senders and receivers are attached to the robot and the target. This generally results in a set of distances, e.g. by measuring the time-of-flight, from which the relative pose can be calculated. Its main advantage is the decrease in computing power compared to image-based techniques. [3] and [4] describe applications in a 2D setting. Figure 2(c) shows an example.

*Email address: maarten.verbandt@student.kuleuven.be

†Email address: bart.theys@kuleuven.be

‡Email address: joris.deschutter@kuleuven.be



(a) Passive optical system: detection of a checkerboard pattern[5]. (b) Active optical system: detection of IR-LEDs[2]. (c) Ultrasound system: intensity measurement[3].

Figure 2: Examples of different measurement systems to dock mobile robots.

This paper describes the development of a robust vision-based measurement system to make a quadrotor land autonomously on a target location, under varying ambient lighting conditions.

2 MARKER DETECTION

A passive optical system is preferable in this case, since it allows for a stand-alone sensor which can be easily attached to any UAV. The proposed marker is based on circles, because they carry positional information in a very natural way, namely by their center-coordinates. In order to improve robustness and to be able to detect the marker from closely, a series of concentric circles with exponentially distributed radii is considered. This assures a similar pattern for varying viewing distance. A multitude of circles also increases the amount of information indicating the marker's center. The proposed marker is shown in figure 3. These circles can be detected by the Hough transform[6] (HT), which is known as one of the most robust detection algorithms in computer vision and will be the core of the detection algorithm described in this paper.



Figure 3: Marker for the developed positioning system.

2.1 Edge detection

The computation of the HT relies on the edges, detected in an image, in which the specified shape is sought. This edge information is extracted by first converting the grayscale image to a binary form. The threshold separating dark and bright pixels is calculated locally with Otsu's algorithm[7], in order to cope with non-uniform lighting conditions. Since the marker contains only very strong edges, crucial information is preserved while weaker edges disappear. The latter increases the signal-to-noise ratio. Additionally, thresholding lowers the required memory to store the image and speeds up the edge detection as it is implemented by logical operations



(a) Image from a scene with the marker. (b) Black-and-white image after local thresholding of 4(a). (c) Edges and gradients extracted from respectively (4(b)) and (4(a)). Only 10% of the gradients is shown.

Figure 4: Graphical illustration of the edge and gradient extraction as part of the localization algorithm.

only. The binary image, resulting from 4(a), is shown in figure 4(b). Edges are detected as black-white transitions.

The calculation of the HT can be improved if also the orientation of the edges is available. This information is captured by the gradient of the grayscale image at the edge's location. Sobel's masks are applied to this extent. Such an implementation is referred to as a generalized Hough transform[8] (GHT). Figure 4(c) shows the extracted edges (crosses) and part of the gradients (arrows).

2.2 Reduced Hough Transform

The basic idea of the generalized Hough Transform is combining the position and gradient information of an edge. This yields a remaining set of possible parameter combinations, describing the particular shape. An accumulation array stores how many times each combination is suggested by one of the edge points. Local peaks in the accumulation array indicate a high probability of the presence of the searched shape with that specific combination of parameters.

The equations of a circle and its gradient are given by respectively (1) and (2). (x_c, y_c) denotes the circle's center, (x, y) one of the extracted edge points and R the radius. These equations can be rewritten in a more useful form: (3) and (4). For any possible radius, the corresponding center can be calculated. This forms a line in the parameter space, indicating all combinations of x_c, y_c and R suggested by the specific edge point. Figure 5 illustrates the 3D accumulation array constructed from the edges shown in 4(c). Local maxima emerge where (x_c, y_c, R) corresponds to an actual circle in the image.

$$C \leftrightarrow (x - x_c)^2 + (y - y_c)^2 = R^2 \quad (1)$$

$$m = \frac{dy}{dx} = -\frac{\partial C / \partial x}{\partial C / \partial y} = -\frac{x - x_c}{y - y_c} \quad (2)$$

$$x_c = x + \frac{m}{\sqrt{1 + m^2}} R \quad (3)$$

$$y_c = y - \frac{1}{\sqrt{1 + m^2}} R \quad (4)$$

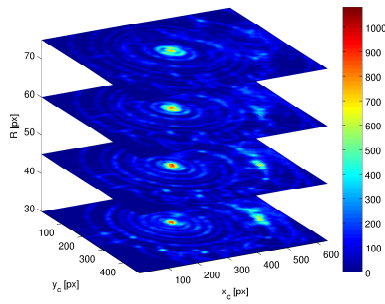


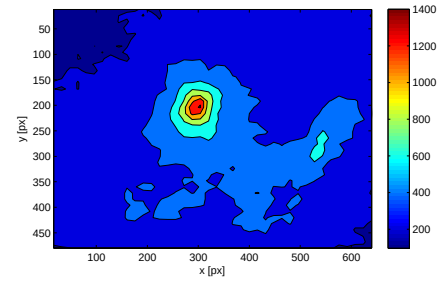
Figure 5: Graphical representation of the accumulation array after the generalized HT (based on the edges from figure 4(c)). Each section corresponding to one of the actual circles has a local maximum at the marker center.

The 3-dimensional parameter space is only useful to extract both the center and radius of all circles. However, localization only involves extracting the center coordinates. Moreover, the circles are concentric which can be exploited to obtain a more robust estimation of the center. This is done by eliminating R from the previous set of equations, giving (5). This implementation is further referred to as the reduced Hough transform.

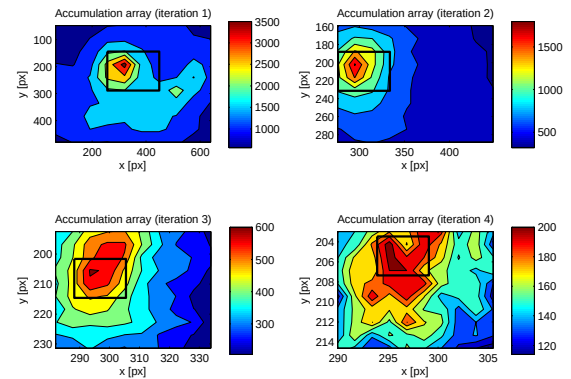
$$L \leftrightarrow x_c = m(y - y_c) + x \quad (5)$$

This way of computing has several advantages. The 3-dimensional parameter space is now reduced to only 2 dimensions, leaving out the unimportant radial information. This not only reduces the required memory but also fuses the position information of the different circles. It also simplifies the actual localization. In case of the standard HT, all peaks require extraction one by one. The reduced HT displays only one dominant peak which is faster and easier to detect. As an example, figure 6(a) presents the 2D accumulation array. The dominant peak in the center indicates the marker. The smaller peak on the right emerges due to a large local concentration of edge points, but is only suggested half as many times.

Because of the reduction from multiple local maxima to only one, an implementation similar to the fast Hough transform[9] (FHT) is very suitable. It iteratively searches for peaks while increasing the amount of detail. This way, only interesting regions are carefully examined. The inherent drawback of the FHT is the need for a threshold, separating useless from interesting regions. Since the marker will cause just one very strong peak, the algorithm should only care about the overall largest value, so the threshold becomes superfluous. This implementation further decreases the required amount of memory and speeds up the detection. The fast implementation is visualized in figure 6(b).



(a) Accumulation array of the reduced Hough transform, based upon the edges from figure 4(c). The central large peak emerges from the marker. The smaller peak on the right side comes from the local excessive amount of non-marker edge points. The latter has only half the magnitude of the central peak.



(b) Iterative implementation of the reduced Hough transform, based on a fast Hough transform implementation. The estimation of the marker position is gradually refined, as indicated by the black boxes.

Figure 6: Graphical illustration of the marker-localization step as part of the localization algorithm.

3 MARKER CLASSIFICATION

In computer vision, classification means labeling the detected objects in an image. In this case, there are only two labels: marker or non-marker. The ability to distinguish between those two labels can be used to filter a badly extracted marker center or verify if a landing platform is present in the scene.

Two distinct measurements (features) are chosen to decide on the eventual label. The first verifies how well the succession of circles matches the expectations. The second compares the amount of retrieved edge points on each circle with the expected amount. Optimally combining these two features is done with a support vector machine (SVM).

Since both features rely strongly on the individual circles, a modified histogram of the radii is constructed. This allows for computing the proposed features, presented in the following 2 sections. An optimal classifier is trained in the last section.

3.1 Histogram of radii

As a first step in extracting the described features, the individual circles are identified. This is done by constructing a histogram of the distances from the edge points to the estimated marker center. Local peaks emerge at distances corresponding to actual circle radii because all points on a circle have the same distance to the center. The histogram of the sample picture (figure 4(a)) is given in figure 7.

One would expect a series of exponentially separated peaks, but this is not the case. There are two main reasons. First, the amount of extracted edge points per circle is proportional to its radius, so the histogram shows a linearly ascending behavior. Dividing each cell by the corresponding radius flattens the histogram, equalizing the height of the peaks. Second, absolute deformation of the observed circles, due to an angle between the marker plane and image plane, is proportional to this angle and also its radius. This deformation is therefore more pronounced for larger circles, affecting the histogram. A suitable way to solve this problem is to use an exponential grid instead of the standard linear grid. The size of each cell increases proportionally to the corresponding radius, allowing more absolute deformation at greater radii. Knowing the ratio of subsequent radii on the marker allows for choosing the base of the exponential grid so that peaks emerge at regular intervals, e.g. one every five cells.

The modified counterpart of the standard histogram is shown in figure 8. The local maxima emerge at regular intervals (one every five cells), are more distinct and easier to extract.

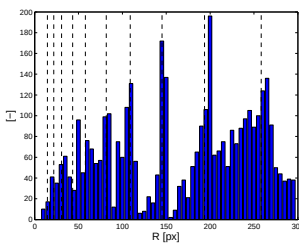


Figure 7: Standard histogram of raw data with linear separation of its cells. The dashed lines indicate the estimated radii.

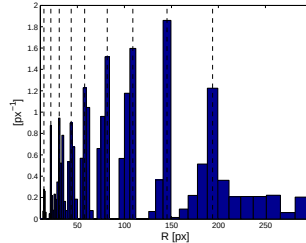


Figure 8: Exponentially separated histogram, including all measures presented in 3.1. The dashed lines indicate the predicted radii.

3.2 Feature 1: circle separation

A first suitable feature for the marker checks the separation between subsequent circles. This is where another advantage of the exponential histogram shows up. Since the amount of cells in between two peaks is fixed, this comes down to measuring the periodicity in the histogram. An excellent tool is the Discrete Fourier Transform (DFT), since periodicity causes a local maximum at the corresponding frequency. However, a direct application of the DFT yields suboptimal results due to aliasing and differing begin and end

values of the histogram.

In order to improve this basic technique, the histogram is cross-correlated with a series of gaussian functions, which separation matches the expected separation of the peaks in the histogram. The resulting sequence has a high value if the gaussians are aligned with the peaks in the histogram. Therefore, it helps distinguishing between the marker and other peaks. The cross-correlation is illustrated in figure 9. Since the periodicity is more outspoken at larger radii, only these are further analyzed (colored red). In order to avoid boundary effects, the sequence is copied and mirrored such that the analyzed signal is smooth. The power spectral density of the periodic sequence is shown in figure 10. It can be seen that more than 50% of the total energy is concentrated at the expected frequency, compared to only 25% when the DFT is applied directly. This power density corresponding to the separation of subsequent circles is used as a first feature: F_1 .

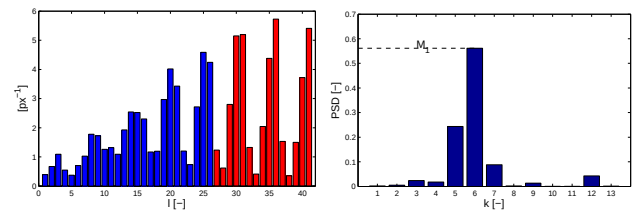


Figure 9: Cross-correlation of the histogram and a series of correlation, rescaled w.r.t. the gaussians. The part with the total power density after removal of the DC component is colored red.

3.3 Feature 2: amount of edge points

The second feature compares the amount of edge points per circle to the expectation. The former is directly retrievable from the histogram, the latter is to be calculated from the circle's radius in the image. In the ideal case, they are equal and their ratio becomes 1. In a real application, this ratio will drop, but it is expected to be larger when an actual marker is detected than in absence of marker. Averaging this ratio over n subsequent circles results in a more stable measurement, hence this average is taken as a second feature: F_2 .

It can be proven this average is proportional to the value of the previously calculated cross-correlation at the index corresponding to radius R . The scaling factor depends solely on the amount of circles included in the average.

3.4 Training the classifier

The previously extracted features are now combined in such a way that the success rate of the classification is maximized. This is done by extracting both features from a set of images containing the marker and a set of random images. Figure 11 shows the distribution of the two classes. It can be seen that the classes are linearly separable, meaning that a

line can be drawn in between the classes (solid line). Of the many classification techniques that exist, a linear support vector machine is most suitable in this case. Here, it comes down to a linear combination of the features. If the resulting value is positive, the object is classified as a marker, otherwise it is not. The optimization problem is formulated such that the majority of samples has a value larger than 1 in case of a marker-image and -1 in case of a non-marker image. This is called the margin and is indicated by the dashed lines. Figure 13 gives an overview of the localization of the marker on different sample pictures of 96x96 pixels, as well as their feature values and the SVM's output. If the latter is positive, the object is classified as a marker. For the images with marker, it can be seen that a good localization ($e_{tot} < 1 px$) yields the correct label. However, the combination of a small marker and bad ambient conditions such as unsharpness and shadow, cause incorrect localization. The extraction is therefore correctly labeled as non-marker. This leaves two mistaken classifications: images 2 and 13, the first being caused by bad lighting, the second by blur. Samples 17 to 20 show how the classifier behaves when the marker is replaced by another round object. The algorithm estimates the marker center to be near the actual circle center. All are classified correctly except image 18. This is because of the high amount of concentric circles makes the object very similar to the actual marker.

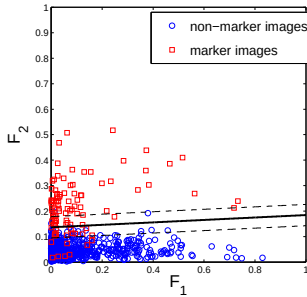


Figure 11: Support Vector Machine, separating the marker and non-marker classes. F_1 and F_2 represent features described in 3.2 and 3.3 resp.

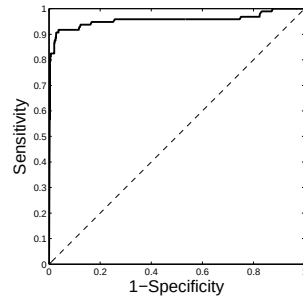


Figure 12: ROC-curve of the Support Vector Machine. The curve is plotted for varying the classification threshold.

4 PERFORMANCE ASSESSMENT

This part verifies the overall performance of the algorithm. This does not only involve tracking accuracy but also robustness against non-uniform lighting and blur, as well as dependence on the size of the marker in the image. Four basic pictures of 480x640 pixels are used, consisting of the marker on a white background. Throughout the images, the size of the marker is varied: 19, 48, 90 and 152 pixels. Shadow and overexposure are mimicked by altering the contrast as well as the average intensity of half of the marker.

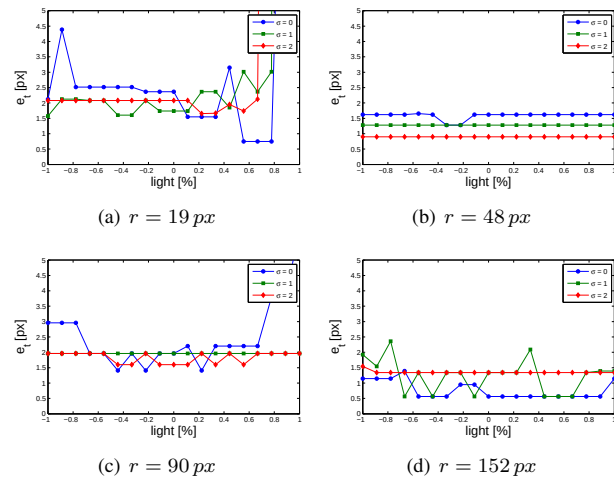


Figure 14: Performance of the localization algorithm under varying lighting, sharpness and distance.

Blur is added by applying a gaussian filter to the images. The resulting degree of unsharpness depends on the variance of the gaussian, σ_G . 19 levels of lighting are treated, going from complete darkness to complete overexposure, along with 3 levels of blurring (σ_G going from 0 to 2), adding up to a total of 228 analyzed artificial images. Examples of these images are found in figure 13 (images 9 to 16).

The presented algorithm is then used to extract the marker from each image. By providing ground truth coordinates for the marker center, the actual accuracy is determined. Figures 14(a) to 14(d) graphically present the results.

The conclusions are threefold. First, the average accuracy appears to be fairly stable when the marker becomes smaller, which indicates a high overall signal-to-noise ratio for the detected marker points. Although objects in a real scene might cause the algorithm to choose a wrong position, the marker will be accurately detected if no other (circular) objects are overwhelming it. Second, moderate non-uniform lighting has little effect, thanks to locally thresholding the image. In the extreme case, where half of the marker is invisible, localization is still acceptable, with total errors ranging from approximately 1 px to 3 px for an average to large marker. However, when the marker becomes too small, the remaining information is insufficient to localize the marker, resulting in large errors. This can be seen in figure 14(a). Third, blur has little effect and can be even advantageous: it stabilizes the outcome of the algorithm. It reduces the high frequency noise, which is desirable in feedback applications.

5 IMPLEMENTATION ON A QUADROTOR

The developed algorithm forms the core of a positioning system, used for automatic docking of a quadrotor. As a quadrotor frame, the DJI f450 is used, which is fur-

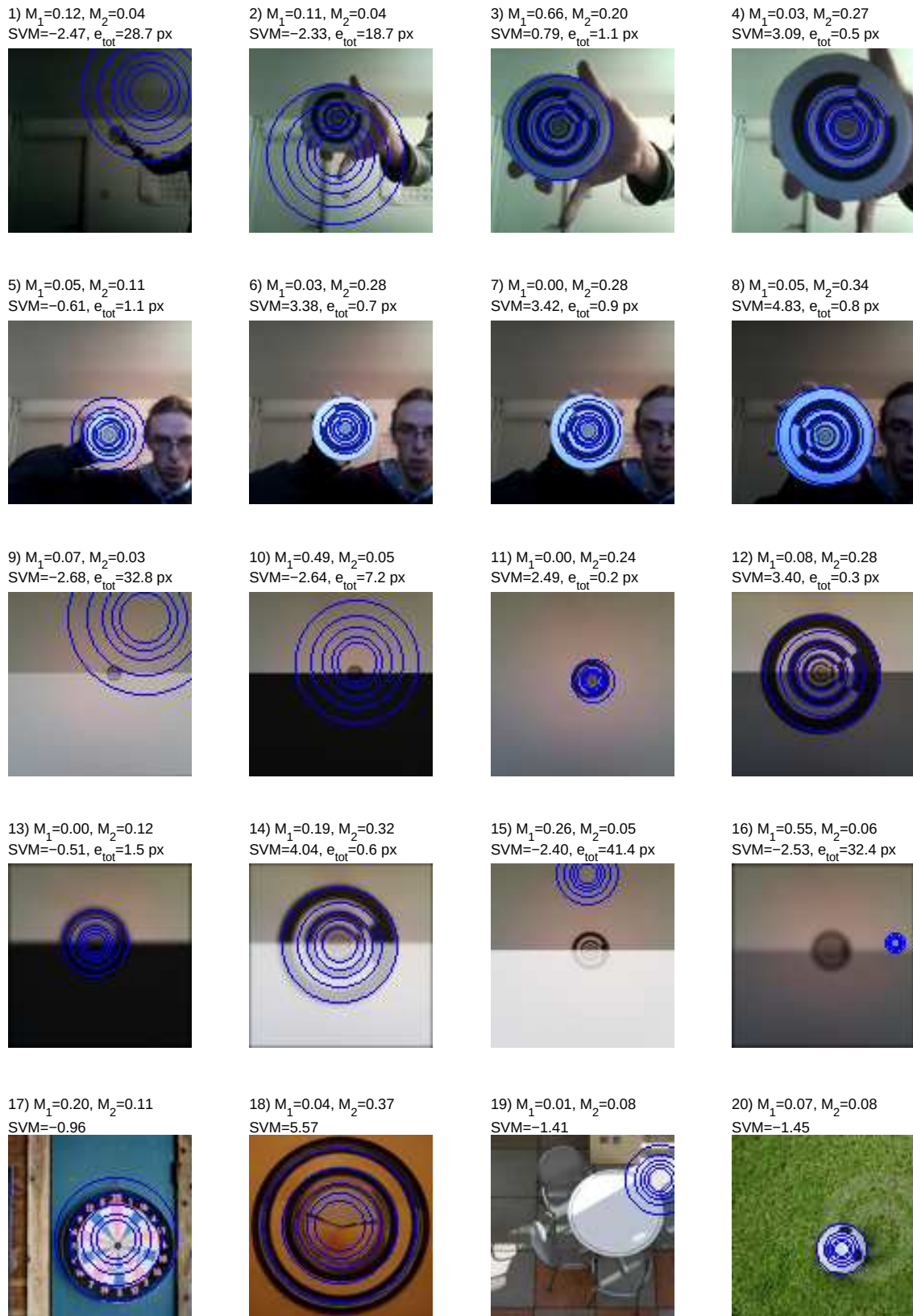


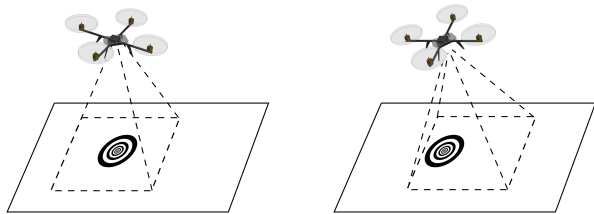
Figure 13: Classification of various images, with and without markers.

ther equipped with the PX4FLOW camera. The camera comes with an ultrasonic distance sensor, which is used to accurately measure the quadrotor's height up to two meters. PX4FLOW is also provided with an ARM processor which is programmed to run an optical flow algorithm. By reprogramming it to execute the marker detection algorithm, a stand-alone positioning module is created. The module can be easily transferred to any other type of VTOL UAV and transmits estimates of the marker's position in image coordinates at 25 Hz.

Ardupilot 2.5 is used as autopilot. It is equipped with an integrated barometer which is used to estimate height when the ultrasonic distance sensor is out of range. Via an i²c connection, the raw position data are read from the PX4FLOW unit and further processed.

The quadrotor arrives at its destination with a base in its viewing range. The algorithm detects the marker, allowing the quadrotor to estimate its position and use it to land on the target.

However, as depicted in figure 15, altering the quadrotor's angles in space causes the marker to shift in the image, while the quadrotor is at the same position. Also changes in the focal length and the distance to the object have to be taken into account. This is referred to as the calibration of the camera and makes it possible to derive a 3D position based on the marker's image coordinates and the distance to the ground. This information is used to estimate the velocity and acceleration of the quadrotor w.r.t. the base which is later used to control the quadrotor's position. Both calibration of the camera and control of the quadrotor are carried out by the autopilot.



(a) Quadrotor with zero roll and pitch angles. The camera points straight to the ground, resulting in an image as indicated by the dashed square. (b) Quadrotor with non-zero roll and pitch angles. The base is shifted in the image compared to figure 15(a), but the quadrotor's position in space remains the same.

Figure 15: Illustration of the need for compensation for non-zero roll, pitch and yaw angles.

5.1 Calibration

Camera calibration is generally split up in internal and external calibration. The internal calibration relies on camera parameters only. Modern digital cameras are usually sufficiently accurately described by a simple model, containing

coordinates of the principal point and pixel mapping parameters. The transformation from image plane to camera coordinates is given in equation 6. ${}^c\vec{p}^{c,m}$ denotes a vector $\vec{p}$, pointing from the camera, c , to the marker, m , expressed in the camera's coordinate system. K represents the internal calibration matrix[10] and maps individual pixels to world coordinates. Since the system is underdetermined, it requires some distance measure to be known in advance, usually the distance to the object d .

$${}^c\vec{p}^{c,m} = d K^{-1} \vec{u} \quad (6)$$

The external calibration compensates for the camera's position w.r.t. the quadrotor's center of gravity, as well as the quadrotor's angles in space. The previously calculated coordinates in the camera frame are subsequently transformed to a stabilized quadrotor frame and fixed world frame.

The orientation of the stabilized quadrotor frame differs from the world frame only by a yaw angle. Computing the relative position of the marker therefore involves compensating for the roll and pitch angles as well as the relative positioning w.r.t. the camera. The transformation is captured by equation 7. ${}^c\vec{p}^{c,q}$ denotes the position vector from the camera to the quadrotor's center of gravity, expressed in the camera coordinate system. ${}^c_qR$ is the rotation matrix, transforming a vector expressed in $\{c\}$ to $\{q\}$. It depends solely on the quadrotor's roll and pitch angles. They are also important for the basic angular control of the quadrotor and are already being calculated from the raw inertial measurement unit (IMU) data.

$${}^q\vec{p}^{q,m} = {}^c_qR ({}^c\vec{p}^{c,m} - {}^c\vec{p}^{c,q}) \quad (7)$$

Transforming the position from quadrotor to world coordinates only requires compensating for the yaw angle. This is expressed in equation 8. ${}^q_wR$ is the rotation matrix taking the yaw angle into account. The yaw angle is measured by the onboard magnetic compass. Considering control in the world frame, it is more suitable to express the base-to-quadrotor position, rather than the other way around, hence the minus in (8).

$${}^w\vec{p}^{w,q} = -{}^q_wR {}^q\vec{p}^{q,m} \quad (8)$$

5.2 Filtering

Directly providing the measured position as feedback to the position controller is not feasible. This is because the marker position is determined by accumulating cells in the image plane which inherently causes high frequency quantization noise. State estimation is therefore necessary to have stable estimates of the velocity and acceleration. To this purpose, a Kalman estimator is used.

In outdoor applications, wind will exert a secondary force on the quadrotor. The positioning system enables the Kalman filter to estimate the equivalent acceleration. This is done by adding this acceleration to the state-space of the quadrotor.

5.3 Preliminary results

Preliminary flight tests show that the measurement system enables the quadrotor to hold its position within a 10 cm radius of the position setpoint (figure 16) at a height of 1 m. Note that the position measurements are taken directly from the quadrotor. In another experiment, a fan exerts an external force on the quadrotor. Figure 17 shows the external force, estimated by the Kalman filter. This force is compensated for by the controller so stable positioning is possible in windy conditions.

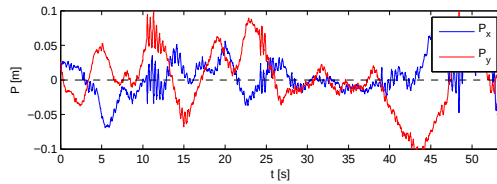


Figure 16: Positioning accuracy of the quadrotor, measured with the developed positioning system at a height of 1 m.

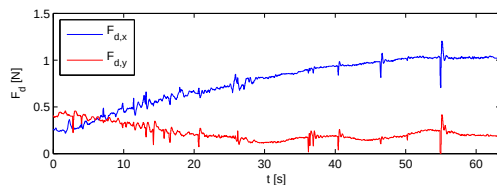


Figure 17: Estimation of the external disturbance of a fan, aligned with the world frame's x -axis. The estimate of the force builds up slowly to a steady value of 1 N.

During a final experiment, the quadrotor carries out six consecutive indoor landings. The eventual positioning errors are shown in table 5.3. The positioning system enables the UAV to land within a 5 cm radius of the target.

Landing	1	2	3	4	5	6
e_x	2.18	3.12	0.76	0.00	0.66	-1.23
e_y	2.93	-0.85	-3.79	1.80	-3.31	-4.07
e_t	3.65	3.24	3.86	1.80	3.38	4.25

Table 1: Position errors after six successive indoor landings in cm.

6 CONCLUSION

The presented vision-based localization algorithm is able to accurately extract the newly developed marker from an image. Local thresholding and the application of an adapted Hough transform both add robustness to the algorithm. The repetitive structure allows detection of both small and large markers in the image, which is of great importance during landing. Also robustness against blurring and non-uniform

lighting is proven. These properties make the algorithm especially suited for use outdoors, where varying lighting is one of the main issues. Apart from localization, validation of the target is also taken into account. This enables the UAV to assess whether a landing platform is present or not.

The developed algorithm is implemented on a stand-alone module that can be easily transferred to any other type of VTOL UAV and transmits estimates of the marker's position in image coordinates at 25Hz. This new position measurement system allows a quadrotor to autonomously maintain a steady position and land within a 5 cm radius of the target. Knowledge of the position makes it also possible to actively estimate and counteract the external disturbances.

REFERENCES

- [1] Bernard Michini, Tuna Toksoz, Joshua Redding, Matthew Michini, Jonathan How, Matthew Vavrina, and John Vian. Automated battery swap and recharge to enable persistent uav missions. *Proceedings of the Infotech@Aerospace 2011*.
- [2] Karl Engelbert Wenzel, Andreas Masselli, and Andreas Zell. Automatic take off, tracking and landing of a miniature uav on a moving carrier vehicle. *Journal of Intelligent and Robotic Systems*, 61(1-4):221–238, 2011.
- [3] Wei Wang, Zongliang Li, Wenpeng Yu, and Jianwei Zhang. An autonomous docking method based on ultrasonic sensors for self-reconfigurable mobile robot. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pages 1744–1749. IEEE, 2009.
- [4] F Tong, SK Tso, and TZ Xu. A high precision ultrasonic docking system used for automatic guided vehicle. *Sensors and Actuators A: Physical*, 118(2):183–189, 2005.
- [5] Companionable - mobile robot companion. smart home. <http://www.companionable.net/>, october 2013.
- [6] Paul Hough. Method and means for recognizing complex patterns, 1962.
- [7] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11:285–296, 1975.
- [8] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.
- [9] Nicolas Guil, Julio Villalba, and Emilio L. Zapata. A fast hough transform for segment detection. *Image Processing, IEEE Transactions on*, 4(11):1541–1548, 1995.
- [10] Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/, april 2014.

Study of design parameters of flapping-wings

Q. Wang*, J.F.L. Goosen, and F. van Keulen

Structural Optimization & Mechanics Group

Faculty of Mechanical, Maritime and Materials Engineering

Delft University of Technology, Delft, The Netherlands

ABSTRACT

As one of the most important components of a flapping-wing micro air vehicle (FWMAV), the design of an energy-efficient flapping-wing has been a research interest recently. Research on insect flight from different perspectives has been carried out, mainly with regard to wing morphology, flapping kinematics, and unsteady aerodynamics. However, the link between the wing morphology and kinematics with passive pitching has been neglected in flapping-wing design. To address this, a model based on a quasi-steady aerodynamic model and the passive pitching motion was made. To simplify the model and make optimization more feasible, the wing is modeled as a stiff plate with uniform mass distribution and a torsional spring at the wing root. An optimization is conducted with the objective of minimizing power consumption for hovering flight using the six most influential wing morphological and kinematic parameters as design variables. The sensitivity of lift generation and power consumption to all the parameters is analyzed. Compared to traditional artificial wings with straight leading edges as pitching axis, wings with a part of wing area in front of the pitching axis and smaller aspect ratio are able to perform more energy-efficient hovering flights. Preliminary design suggestions regarding the selection of wing shape and kinematics for FWMAVs are given.

1 INTRODUCTION

Inspired by gliding birds, engineers designed airplanes equipped with fixed wings to fly when moving at a high speed through the air. This design concept has proved successful in long-distance transport, but has bad performance in the low-speed range and is useless for hovering. The helicopter designs solve this with rotary wings providing the possibility to hover and move slowly. However, when we reduce the size of helicopters to execute tasks in limited space, the noise from fast rotating blades becomes a big issue and the collision with other objects is often fatal. Given aforementioned reasons,

designers switched to mimic the flapping flight of birds and insects since they are the specialists in nature on slow and agile flight especially with small body sizes. Several kinds of flapping-wing micro air vehicles (FWMAVs) [1, 2, 3] were designed by mimicking insect flight, which requires less control of the wing itself compared to bird flight.

As a dominant part for the generation of lift and thrust, study on the flapping-wings of insects has been conducted from perspectives of wing morphology, kinematics, unsteady aerodynamics, etc. Biologists have observed insect flight for centuries and studied the wing morphology a lot in the early days. Ellington [4] studied the relation between the radii of wing area for a variety of insects, birds and bats, and found that wings of flying animals adhere to “laws of shape”, implying the possibility for analytical representation of the spanwise wing area distribution. Wootton [5] elaborately studied and summarized the functionality of insect wing morphology. The combination of wing venation and membrane determines the spanwise stiffness to avoid excessive wing bending but allows a certain spanwise flexibility for wing pitching and torsion. This morphological characteristic facilitates the generation of expected kinematics for energy-efficient flight with the help of wing inertial and aerodynamic loads. Several studies [6, 7, 8] have investigated why insects adopt certain kinematics from the aerodynamic performance and energy efficiency perspectives. Numerous optimization for the flapping-wing kinematics were also used to strengthen the understanding of insect flights [9, 10] and assist the design of FWMAVs [11, 12].

Although research has focused both on wing morphology and flapping kinematics, little work has been carried out to systematically study the morphological and kinematic parameters simultaneously. Insects with different wing morphology generally flap in different ways, which reflects the mutual relationship between wing morphology and kinematics. This can be partly attributed to the fact that most insects make use of passive pitching [13, 14], and the passive pitching is determined by both the wing morphology and flapping motion. Additionally, kinematics with passive pitching are also preferable for FWMAVs since this simplifies the drive mechanism and, thus, reduces the total mass. Thus, it is necessary to do a combined analysis of both wing morphological and flapping kinematic parameters while the pitching motion is passive. As the kinematics are strongly dependent on the flight mode, a specific flight mode must be chosen for the analysis. This paper concentrates on hovering flight for two

*Q.Wang-3@tudelft.nl

reasons. First, insects and FWMVs exhibit a variety of forward flights with different forward speeds, flight trajectories and objectives (e.g., for maximum flying time or distance). This variation makes the forward flight hard to specify for quantitative analysis. In contrast, most hovering flights share the same objective: minimal power consumption (or maximal hovering time). Second, hovering flight is a requisite capability for most insects and FWMVs, and it requires more energy than forward flight in intermediate speed [8], which implies the importance of studying hovering flight to reduce the overall energy expenditure.

In this paper, wing morphological and kinematic parameters are first described in detail. Then, with preliminary analysis of these parameters, the most influential ones are selected as design variables for optimization. Finally, based on the optimal combination of wing shape and kinematics, single parameter sensitivities of lift and power consumption are calculated and analyzed.

2 DESIGN PARAMETERS

Insect wings are thin cuticular structures that consist of membranous regions of epidermal bilayers reinforced by veins [15]. The veins are used to maintain the wing shape and facilitate the aerodynamic loads generation. To this end, they increase the spanwise stiffness to reduce wing bending but allow chordwise flexibility that permits the twisting necessary to achieve the required angles of attack.

Instead of pitching the wing with distributed flexibility, we mimic the passive pitching motion by connecting a rigid wing to a torsional spring, as shown in Figure 1. Ignoring the deformation of the wing itself and, thus, the influence of wing bending, torsion and chamber, the morphological parameters are reduced to pure wing shape parameters.

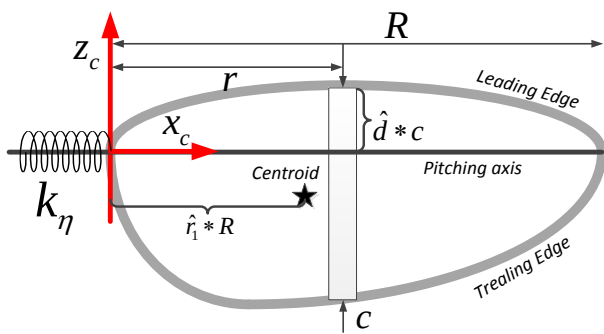


Figure 1: Illustration of the wing model with a detailed description of shape parameters.

2.1 Wing morphological parameters

To quantitatively describe a wing shape, the line from the wing root to tip is defined as a reference line. This line is also assumed to be the pitching axis in the kinematics description.

Next, the leading edge (LE) and trailing edge (TE) can be identified clearly, as shown in Figure 1.

The wing area (S) enclosed by the LE and TE has the most significant influence on the lift and thrust generation, and is conventionally quantified with the wing span (R) and aspect ratio (AR), which is defined as the ratio of R to average chord length ($\bar{c}$) for a single wing. In most cases, R is proportional to insect body length, but AR might range from 1 to 5 for different insect species [15]. Small AR typically implies less maneuverability but smaller wing loading, as displayed by butterflies. Consequently, most insects adopt moderate wing aspect ratios to find a balance between the wing loading and flight agility.

Obviously, it is not possible to fully describe the wing shape merely with R and AR as we can move the wing area longitudinally and transversely. Ellington [4] studied the relation between the radii of wing area for a variety of insects, birds and bats, and proposed to use the Beta probability density function (BPDF) to describe the spanwise area distribution. Two parameters of the BPDF, namely mean value and standard deviation, are equivalent to the non-dimensional first and second radius of moment of area, $\hat{r}_1$ and $\hat{r}_2$. His “laws of wing shape” show that there exists an approximate relationship between $\hat{r}_1$ and $\hat{r}_2$: $\hat{r}_2 = 0.929\hat{r}_1^{0.732}$. Hence, the spanwise area distribution can be characterized by a single parameter $\hat{r}_1$, which also indicates the spanwise centroid position and usually ranges from 0.4 to 0.6 for insect wings. Wang *et al.* [12] studied the effect of $\hat{r}_1$ on the optimal hovering kinematics and flight efficiency, and showed that locating more area towards the wing tip leads to more efficient hovering flight. However, their model ignored the power consumption resulting from the generated side force, which becomes more significant when more area is put close to the wing tip. Consequently, wings with moderate values for $\hat{r}_1$ are likely more energy efficient, for instance, the values of $\hat{r}_1$ for most Diptera insects are between 0.45 and 0.55 [4].

If we look at the insect wing shapes of different species, there are different percentages of wing area distributed in the front of the pitching axis. It is therefore necessary to introduce additional parameters to describe the chordwise area distribution. The non-dimensional length between the LE and pitching axis, $\hat{d}(r)$, which is normalized by the local strip length, is used to characterize the chordwise area distribution. For simplicity, the change of $\hat{d}$ along the span is assumed to be linear, which already enables to represent a huge number of wing shapes. Hence, we introduce shape parameters $\hat{d}_{\text{root}}$ and $\hat{d}_{\text{tip}}$ to define $\hat{d}(r)$.

With fully described wing shape, which is crucial for the passive pitching motion and power consumption, the moment of inertia (I) can be determined if we know the mass distribution over the wing planform. For insect wings, the venation distribution dominates the mass distribution. The enrichment of veins at the wing root and near the pitching axis makes the mass distribution heterogeneous, which decreases the mo-

ment of inertia and, thus, the energy loss due to wing inertia. However, we still lack knowledge on the distribution of the wing mass for an arbitrary wing shape. Consequently, a uniformly distributed wing mass is hereby assumed.

2.2 Flapping kinematic parameters

The most significant characteristics separating the insect flight with fixed-wing and rotary-wing vehicle flight is that flying insects reciprocate their wings to exert moment on the surrounding air to produce lift and thrust. This reciprocating motion leads to a spanwise velocity gradient along the wing and continuously varying translational and rotational velocity during the periodic rotation of the wing. All these characteristics are crucial for the generation of sufficient aerodynamic forces for insects.

Considering the complexity of wing spatial movement, three Euler angles are generally used to describe the flapping kinematics. They are sweeping angle ϕ (yaw), heaving angle θ (roll) and pitching angle η (pitch), as illustrated in Figure 2. Two coordinate systems (CSs) are of particular interest for the study of flapping wing motion, i.e. the inertial CS $x_i y_i z_i$ and the co-rotating CS $x_c y_c z_c$, which are fixed on the body and wing, respectively. Both CSs are shown in Figure 2. Aerodynamic forces are preferably calculated in the co-rotating CS and subsequently transformed into the inertial CS to evaluate whether the wing design is able to produce sufficient lift and thrust or not. The power consumption is also calculated in the co-rotating CS since the wing moment of inertia does not change with wing flapping.

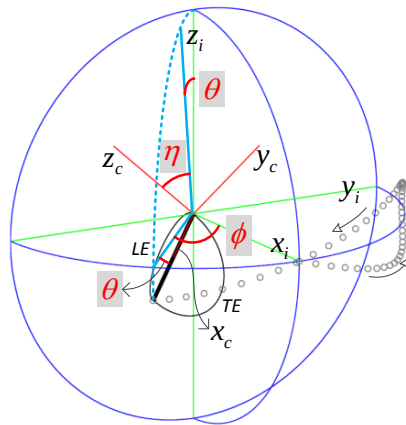


Figure 2: Definition of Euler angles for studying flapping-wing kinematics.

In nature, a variety of wing tip trajectories can be found, and a collection was done by Berman and Wang [10]. Most wing tip trajectories can be categorized as three types, i.e., "∞" shape, "∩" or "∪" shape and "—" shape. The shape of "—" implies that no heaving motion is existent, and it is the most frequently used kinematic pattern by FWMAVs. In or-

der to quantitatively describe these tip trajectories, the mathematical representation with three Euler angles is necessary.

The wingbeat frequency (f) of different flying insects exhibits huge differences, ranging over three orders of magnitude from 5.5 Hz (Damselfly) to 1046 Hz (Chironomoidea) [15]. Generally, f varies inversely with wing inertial loads [16], especially for those insects exploiting resonant systems to flap their wings. That is coincident with our intuitive feeling about flying insects. For instance, larger insects usually flap their wings more slowly, and insects with small aspect ratio wings (e.g., butterfly) also fly with relatively low f as compared to those with comparative body mass and wing span. Additionally, if the flying circumstance does not vary a lot, the flapping frequencies of individual insect species are fairly constant, and their variation coefficients are within 5% for insects in hovering flight as studied by Ellington [17]. This can be partly explained by the exploitation of the resonant drive system by insects. Insects prefer to flap their wings at the resonant frequency to save energy.

Insects reciprocate their wings with quite different stroke amplitudes, which could range from 66° for the Syrphid fly [17] to 180° or even higher for some beetles and moths [15]. Several studies [10, 12] show that higher stroke amplitude is beneficial for a more efficient flight. However, some limitations constrain the stroke amplitude to achieve or exceed 180° . These constraints include the geometric constraint due to wing-body interference and physical constraint of the wing-driving apparatus of insects or FWMAVs. The sweeping motions of insect flight are usually in a sinusoidal or near sinusoidal pattern, while some flapping-wing drive mechanisms [18, 19] generate sweeping motions in a triangular pattern. The triangular sweeping motions experience quick reversals between half-strokes and fairly constant wing translational velocities. Taha *et al.* [20] found that a triangular waveform sweeping angle and a constant pitching angle throughout the half-stroke yield the optimal performance while only aerodynamic power is considered. However, that specific pitching motion is hard to generate. The kinematics of some insect flights also shows an asymmetric downstroke and upstroke, but symmetric sweeping motion is more general in hovering flights of insects and FWMAVs. In order to generate a pitching moment for flight control, some insects [7] and FWMAVs [21] adopt offset sweeping motions, where the centerline of the sweeping motion angles to the front or rear. Here, we use the model proposed by Berman and Wang [10] to describe the sweeping motion for hovering flight:

$$\phi(t) = \frac{\phi_m}{\arcsin K} \arcsin [K \sin(2\pi ft)] + \phi_0, \quad (1)$$

where ϕ_m and ϕ_0 are the sweeping amplitude and horizontal offset, respectively, and K ($0 < K < 1$) is a parameter used to control the shape of the sweeping motion: if $K \rightarrow 0$, $\phi(t)$ becomes a sinusoidal pattern, and if $K \rightarrow 1$, $\phi(t)$ becomes triangular.

To describe the out-of-plane movement of flapping wings, the heaving angle θ is described by a sinusoidal function,

$$\theta(t) = \theta_m \sin(2\pi Nft + \Phi_0) + \theta_0, \quad (2)$$

where N takes either 1 for the "∩" or "∪" shape or 2 for the "∞" shape, and θ_m , Φ_0 and θ_0 are the sweeping amplitude, vertical phase shift and vertical offset, respectively.

Pitching angle (η) provides the information of the geometric angle of attack (AOA) (α_{geo}), e.g., $\alpha_{\text{geo}} = 90^\circ - |\eta|$ if there exists no heaving motion. Together with the induced velocity and translational forward velocity, the effective AOA (α_{eff}) can be obtained. The pitching angle demonstrated in insect's flight mainly originates from the wing deformation, which is a combination of wing bending, camber and torsion. This deformation is determined by distributed wing stiffness and wing inertia as well as aerodynamic loads. In this paper, the distributed wing stiffness is replaced by the rotational stiffness of the torsional spring based on the rigid wing model. Since the wing inertia can be calculated easily in the co-rotating CS, the calculation of transient aerodynamic loads is the key to simulate the passive pitching motion.

A quasi-steady aerodynamic model is used to calculate the transient aerodynamic loads. The loads are decomposed into four components that originate from different sources: wing translational velocity, rotational velocity, coupling between the wing translational and rotational velocities, and added mass effect, as illustrated in Figure 3. Specifically, we use an analytical model proposed by Taha *et al.* [22] to calculate the loads due to the wing translational velocity. The analytical model enables the calculation of the lift coefficient for a translational 3-D wing with arbitrary aspect ratio, which facilitates the aerodynamic analysis of different wing shapes. By contrast, previous quasi-steady models [23, 24] estimated the wing translational loads with empirical data from experimental measurements on specific insect-wing-like mechanical wings, which restricts its application to different wing shapes. Although the wing rotational velocity does not contribute to the average forces over a cycle with two symmetric half-strokes, it is an important part of the transient damping torque around the pitching axis and, thus, included in our model. If the pitching axis is located at the chord center, no net transient force results from the wing rotational velocity. However, there still exists an additional force, which is a result of the coupling effect between the wing translational and rotational velocities [23, 25]. An analytical formula proposed by Fung [26] for an oscillating plate with small AOA shows a good prediction of the force due to the coupling and, thus, is used in our model. The center of pressure due to the coupling is at the 3/4 chord length from LE. The quasi-steady model including the damping torque from this term shows a better agreement with measured the passive pitching motion in experiments than the model without this term [27]. The added mass effect is due to the wing acceleration or deceleration, which leads the surrounding air to be accelerated or deceler-

ated as well. The disturbed air will, in turn, exert forces on the wing.

Because of the variation in the velocity and acceleration along the wing span, the blade-element method [28], which discretizes the wing into infinitesimal chordwise strips, is used together with the quasi-steady model. The loads in the co-rotating CS can be calculated by integrating the loads on each of the strips over the entire wing. The average loads can be obtained by averaging the transient forces over a stroke cycle.

According to Euler's second law of motion for a rigid body, the external torque about the axis x_c is equal to the time derivative of angular momentum about the same axis,

$$\tau_{x_c}^{\text{aero}} + \tau_{x_c}^{\text{spring}} = \frac{dL_{x_c}}{dt}, \quad (3)$$

where L_{x_c} is the x_c component of angular momentum $\mathbf{L}$ of the wing expressed in the co-rotating CS, which is equal to $\mathbf{I}\boldsymbol{\omega}_c$, where $\boldsymbol{\omega}_c$ represents the angular velocity of the wing in the co-rotating CS. Then, the equation of motion for the passive wing pitching can be derived as:

$$\begin{aligned} I_{x_c x_c} \ddot{\eta} = & -k_\eta \eta + \tau_{x_c}^{\text{aero}} + I_{x_c x_c} \left[\frac{1}{2} \dot{\phi}^2 \cos^2 \theta \sin(2\eta) \right. \\ & - \frac{1}{2} \dot{\theta}^2 \sin(2\eta) + 2\dot{\phi}\dot{\theta} \cos \theta \cos^2 \eta + \ddot{\phi} \sin \theta \left. \right] \\ & + I_{x_c z_c} [\ddot{\theta} \sin \eta + \frac{1}{2} \dot{\phi}^2 \sin(2\theta) \sin \eta \\ & - \ddot{\phi} \cos \theta \cos \eta + 2\dot{\phi}\dot{\theta} \sin \theta \cos \eta], \end{aligned} \quad (4)$$

where $\tau_{x_c}^{\text{aero}}$ is an implicit function of η , $\dot{\eta}$ and $\ddot{\eta}$. Then, the pitching angle can be obtained by numerically solving this second order ordinary differential equation. Using the quasi-steady aerodynamic model and the resulting kinematics, an analysis of the effects of the parameters on the lift generation and power consumption can be conducted.

3 ANALYSIS OF DESIGN PARAMETERS

3.1 Preliminary analysis

The majority of parameters discussed above have been studied from the lift and thrust generation perspectives [5, 10, 12, 29, 30]. For instance, longer wing span, higher flapping frequency and larger sweeping amplitude tend to generate more lift force. A medium value of the pitching amplitude ($40^\circ - 60^\circ$) also benefits the lift generation. Heaving motion is likely to enhance lift generation, since the plunging after wing reversals can increase the effective AOA. However, the influence of these shape and kinematics parameters on the power efficiency is not clear. Therefore, it is necessary to explore how to select these parameters for a more efficient flight, which is important for designing FWMAVs.

Power consumed by flapping flight mainly consists of four parts: profile power (P_{pro}), parasitic power (P_{par}), induced power (P_{ind}) and inertial power (P_{iner}). The first three can be categorized as aerodynamic power (P_{aero}). Specifically, P_{pro} results from the work done to overcome the drag

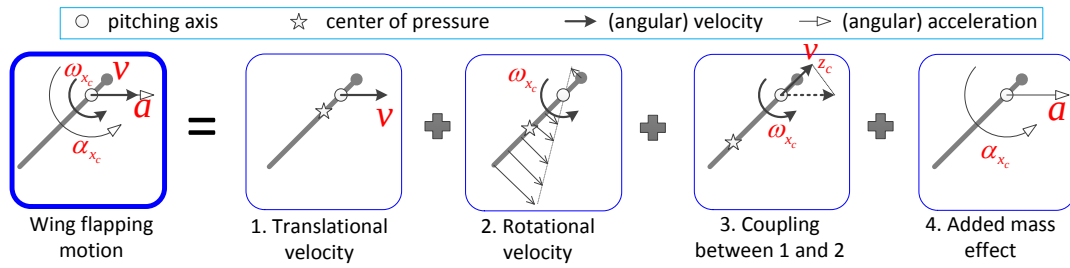


Figure 3: Decomposition of the quasi-steady aerodynamic modeling. The gray line and filled circle represent the wing chords and leading edge, respectively.

Parameters	Descriptions	Design domain in optimization	Ranges of parameters in sensitivity analysis
$\hat{r}_1$	normalized radius of first moment of area	0.45	[0.4, 0.6]
R [m]	wing span	0.05	[0.045, 0.055]
AR^*	aspect ratio	[1, 4]	[1, 4]
$\hat{d}_{root}^*$	$\hat{d}$ at the wing root	[0, 0.5]	[0, 0.5]
$\hat{d}_{tip}^*$	$\hat{d}$ at the wing tip	[0, 0.5]	[0, 0.5]
k_η [Nm/rad]*	rotational stiffness	$10^{-4} \times [1, 15]$	$10^{-4} \times [1, 15]$
f [Hz]*	flapping frequency	[5, 30]	[5, 30]
ϕ_m [rad]	sweeping amplitude	$\{\pi/4, \pi/3, \pi/2\}$	$\{\pi/4, \pi/3, \pi/2\}$
K	control sweeping pattern	0.01	[0.01, 0.99]
ϕ_0 [rad]	horizontal offset	0	$[-\pi/30, \pi/30]$
θ_m [rad]	heaving amplitude	0	$[0, \pi/6]$
Φ_θ [rad]	heaving phase shift	0	$[-\pi, \pi]$
θ_0 [rad]	vertical offset	0	$[-\pi/10, \pi/10]$
N	control wing tip trajectory	0	2

Table 1: List of parameters, their domains in optimization and sensitivity analysis. Parameters labeled with “*” are selected as design variables in optimization. The design domains in optimization refer to the frequently studied insects (moths, beetles) and FWMAVs.

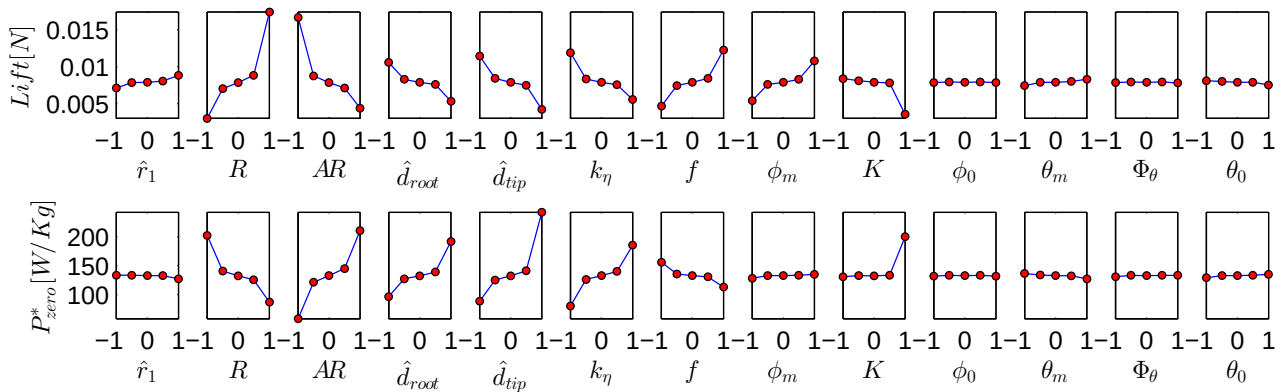


Figure 4: Importance comparison of design parameters in response to lift and power consumption P_{zero}^* . The values “-1” and “1” represent the selected lower and higher setting of each design parameters. These filled circles in red indicate the mean lift or power consumption of all the trials while fixing the corresponding parameter at certain level.

on the wing. P_{par} is the power required to overcome drag on the body, which is negligible for hovering flight but significant for high-speed forward flight. P_{ind} is expended to overcome the drag on the wing due to induced downwash velocity which was estimated by the (partial) actuator-disk model [31]. However, this model does not consider any unsteady aerodynamic effects, which makes it not accurate to estimate the P_{ind} . Nevertheless, P_{ind} is generally less than 20% of the total power [32] and does not vary a lot with the wing morphology and kinematics, except for the stroke amplitude. Thus, P_{ind} is ignored in the analysis of the effect of these design parameters on power efficiency. Additional to aerodynamic power expenditure, P_{iner} is required to accelerate the wing mass and virtual mass due to accelerated surrounding air in the first half of a half-stroke. In the second half, the decelerating wing will lose its kinetic energy, which might be used to overcome aerodynamic drag or stored in the cuticle, flight muscles and resilin for insects [15] or elastic structure of FWMAVs [2]. This kinetic energy might also be dissipated by the thorax structure, which makes the power efficiency of flapping flight hard to determine. In this paper, two patterns of power consumption are studied:

- Perfect energy storage, which means all kinetic energy can be either used to compensate the power consumed by the drag or recycled by the drive mechanism, termed as P_{per} .
- Zero energy storage but perfect energy transfer between inertial and aerodynamic work, which implies that if stored kinetic energy in the first half of a half-stroke is less than P_{aero} required for the second half of the half-stroke, all the inertial power is fully recycled and total power is equal to P_{per} , otherwise, part of the kinetic energy will be dissipated. It is denoted as P_{zero} .

The power consumption of flapping flight generally lies between P_{per} and P_{zero} . To facilitate the comparison of energy-efficiency between different flapping-wings, the power consumption is normalized by the transient lift (in kilograms, if negative taking the absolute value) and denoted with an asterisk, e.g., normalized aerodynamic power consumption P_{aero}^* .

In Table 1, all the design parameters are listed. To gain insight into the importance of these parameters to the lift generation and power consumption, the screening of these parameters with 5-level fractional factorial experimental design approach [33] is first carried out. It can be seen from Figure 4 that: R , AR , $\hat{d}_{\text{root}}$, $\hat{d}_{\text{tip}}$, f and k_{η} have more influence on both lift generation and power consumption than $\hat{r}_1$, ϕ_0 and all the heaving motion parameters. ϕ_m shows relatively more influence on lift generation than on power consumption. Although the interaction and higher-order terms of these parameters are not included in the screening, the results provide a rough picture of parameters that should be included in the design variables in optimization.

Table 2: Optimization results.

ϕ_m	$\pi/4$	$\pi/4$	$\pi/3$	$\pi/3$	$\pi/2$	$\pi/2$
P_{per}^*	16.46	—	18.89	—	23.40	—
P_{zero}^*	—	21.53	—	23.93	—	27.70
AR	1.00	1.00	1.00	1.00	1.00	1.00
$\hat{d}_{\text{root}}$	0.12	0.26	0.12	0.14	0.18	0.24
$\hat{d}_{\text{tip}}$	0.42	0.22	0.48	0.22	0.39	0.08
k_{η} ($\times 10^{-3}$)	1.32	1.15	1.03	1.25	0.82	1.01
f	14.74	13.89	13.19	11.88	10.41	9.80
$\eta_m(^{\circ})$	43.63	49.67	46.69	50.25	52.36	55.62

3.2 Optimization

The optimization is set up to search for the combination of wing shape and flapping kinematics with the least power consumption but with the minimal lift generation (1[g]) for a single wing. To make the optimization meaningful and manageable, some parameters are prescribed similar to the size of an adult Hawk moth, which are frequently used by FWMAVs [2, 34]. R and $\hat{r}_1$ are set as 50 [mm] and 0.45, respectively; the average wing density is set to 0.05 [Kg/m^2]; three discrete values $\pi/4$, $\pi/3$ and $\pi/2$ are considered for ϕ_m of a harmonic sweeping motion with $K = 0.01$; ϕ_0 and the heaving motion are ignored in optimization. For the other five design parameters, i.e., AR , $\hat{d}_{\text{root}}$, $\hat{d}_{\text{tip}}$, k_{η} and f , optimization is carried out to search for the most energy-efficient combination of these parameters corresponding to both power patterns.

Given that solving the ODE (Eq. 4) may have convergence problem and the power defined in the design domain is a non-convex set, a global optimization method is used to search for a rough solution. Next, a response surface is fitted around the rough optimum with using a cubic spline, based on which the gradient-based optimization is carried out for the determination of the optimum.

For a rigid wing with uniform mass distribution, the optimal combinations of wing shape and kinematics for hovering flight without heaving motion are listed in Table 2 with respect to three discrete sweeping amplitudes. From the results it can be concluded that:

- 1 Relative to adult Hawk moths, wings with minimal aspect ratios ($AR = 1$) are obtained, as shown in Figure 5. A smaller AR increases the wing area and, thus, the lift generation. However, the power consumption also increases due to the increase of wing inertia. Both the optimal location of the pitching axis and flapping frequency are used to compensate the increased power consumption due to the smaller AR . The optimal pitching axis is located close to the wing center line and center of pressure, which is located between 1/4 and 1/2 chord length from LE. Meanwhile, smaller flapping frequencies, especially for P_{zero}^* , are also found by optimization. These optimal wing shapes suggest

that having wing area in front of the pitching axis is more energy-efficient than those with straight leading edges as the pitching axis. The latter designs are quite commonly used for artificial wings of FWMVs.

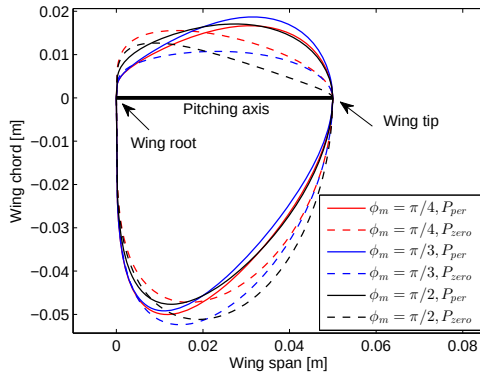


Figure 5: Optimal wing shape for both power patterns with three different sweeping amplitudes.

- 2 The optimal flapping frequencies are inversely proportional to the sweeping amplitudes. The power consumption with respect to the optimal design increases with the sweeping amplitude when the induced power is ignored. This conclusion is different from that obtained for the flapping motion with fully active kinematics [10, 12], which shows that a larger sweeping amplitude is more energy efficient. To achieve the optimal pitching amplitudes (η_m), as listed in the last row of Table 2, the spring stiffness (k_η) is tuned to different values.

3.3 Sensitivity analysis

Sensitivities of the lift and power consumption for single parameters for both power patterns, with $\phi_m = \pi/3$, are plotted in Figure 6 and 7. The sensitivities plotted in subplots from (a) to (i) are calculated based on the prescribed values of $\hat{r}_1$, R , K and ϕ_0 , and the optimal values of five design variables in the optimization without considering the heaving motion.

For both power patterns, an increase of $\hat{r}_1$ or R does not always generate higher lift and reduce energy efficiency if the pitching axis and rotational stiffness are fixed, since there exists an optimal moment of inertia that is related to $\hat{r}_1$ or R which gives rise to the preferable passive pitching amplitude. According to sensitivities in subplots from (c) to (g) for both power patterns, the optimal design obtained with the objective of the minimal power consumption are approximately coincident with the optimal design for the maximal lift generation. With the increase of parameter K , as shown in subplot (h), both the lift and mass-normalized power decrease for the power pattern P_{per}^* , but for P_{zero}^* the power decreases between 0.01 and 0.9 and then increases due to high wing inertia at the

fast wing reversal phase. From subplot (i) in both figures, it can be found that the horizontal offset of the stroke plane has no influence on the aerodynamic force generation or power consumption. However, the location where the average lift is located with respect to the center of mass will move and, thus, the moment around the center of mass will change. This is useful for flight control.

To study the effect of heaving motion parameters on the lift and power consumption, first, the heaving amplitude θ_m is varied between 0 and $\pi/6$. Then, by setting θ_m as $\pi/10$, the sensitivities with respect to Φ_θ and θ_0 are calculated and plotted in subplots (k) and (l). It can be seen that hovering performance can be improve by turning these three parameters, although a more complicated drive system is required to achieve the desired heaving motion.

The peaks of power consumption in subplots (f) and (g), which are no shown, are due to the absolute value of transient lift, which is used to normalize the power, getting close to zero. For other cases with different sweeping amplitudes, the results of sensitivities are not shown as no significant difference with Figures 6 and 7 is found.

4 CONCLUSIONS

Passive pitching motion is preferable for flapping wing design since it simplifies the drive mechanism, saves energy and reduces structural weight. The passive pitching is related to both the wing shape and flapping kinematics, which requires the combined design of these two aspects. To obtain an insight into the effect of these parameters on lift and power consumption, five shape parameters and seven kinematics parameters as well the stiffness of the torsional spring have been studied. The passive pitching motion is simulated based on a rigid wing model and a quasi-steady aerodynamic model. Based on the importance analysis of these parameters, five parameters, including AR , $\hat{d}_{root}$, $\hat{d}_{tip}$, k_η and f , are selected as design variables in optimization while other parameters are prescribed.

Optimization results show that smaller aspect ratio wings with the pitching axis located close to the center of pressure and wing center line are more energy-efficient regardless of the elastic energy storage. Smaller power consumption can be achieved by the optimal wing design with smaller sweeping amplitude. Sensitivity analysis to all the design parameters shows the effect of these parameters on the lift and power consumption, which can assist in the selection of wing shape and kinematics pattern and help determine effective control parameters using the lift sensitivities.

This work is based on the assumption of a rigid wing model with uniformly distributed mass over the wing surface, which facilitates the comparison of different wing designs. Although this simplified model can not completely characterize the complexed morphology of insect wings, the optimal wing shapes and kinematics show the dependence of the performance on the moment of inertia of the wings and can be

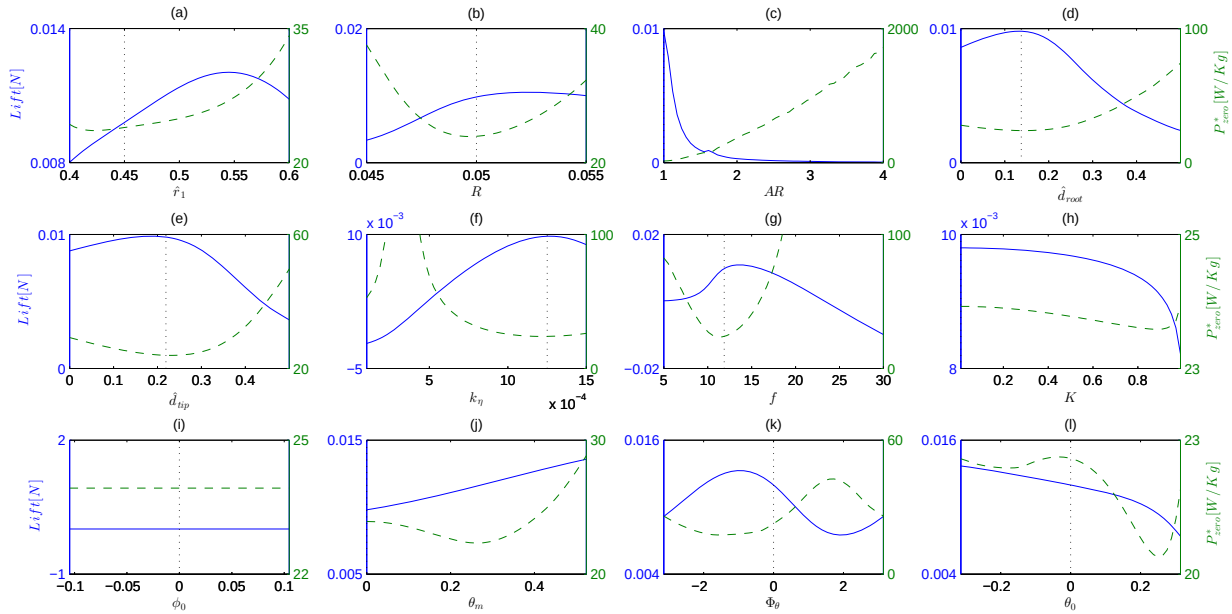


Figure 6: Effect of single parameters on the lift (solid line) and the power pattern P^* (dashed line) while $\phi_m = \pi/3$. The values on x axes corresponding to dotted lines are either the optima from optimization (subplots (c)-(g)) or the prescribed values in optimization in other subplots. The dotted lines in subplots (k) and (j) overlap with left vertical axes. The sensitivities with respect to Φ_θ and θ_0 are calculated while set θ_m as $\pi/10$.

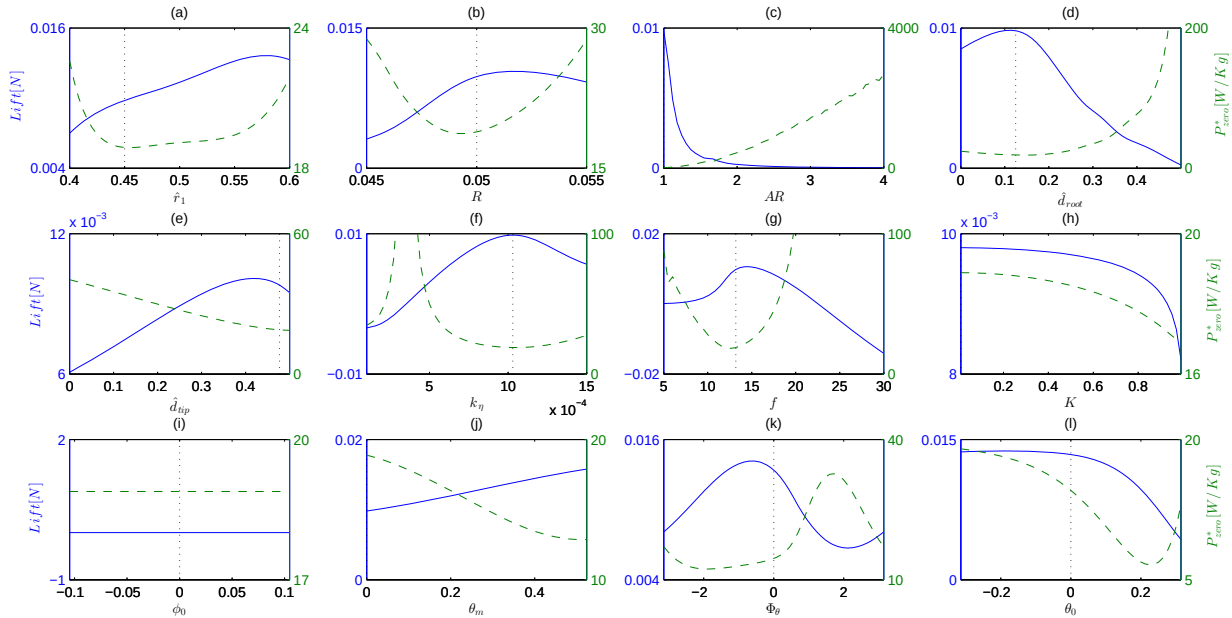


Figure 7: Effect of single parameters on the power pattern P^* while $\phi_m = \pi/3$. Details as in Figure 6.

used in the FWMAV design. In future, it is helpful to include forward flight and maneuvering as well to achieve more comprehensive understanding of these design parameters on the flight performance. In addition, wing torsion resulting from distributed wing stiffness is likely to further reduce the power consumption and should be considered in future modeling as well.

ACKNOWLEDGEMENTS

We would like to thank the China Scholarship Council for their financial support.

REFERENCES

- [1] G. C. H. E. de Croon, K. M. E. de Clercq, R. Ruijsink, B. Remes, and C. de Wagter. Design, aerodynamics, and vision-based control of the DelFly. *International Journal of Micro Air Vehicles*, 1(2):71–98, 2009.
- [2] C. T. Bolsman, J. F. L. Goosen, and F. van Keulen. Design Overview of a Resonant Wing Actuation Mechanism for Application in Flapping Wing MAVs. *International Journal of Micro Air Vehicles*, 1(4):263–272, December 2009.
- [3] 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, May 2013.
- [4] C. P. Ellington. The Aerodynamics of Hovering Insect Flight. II. Morphological Parameters. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 305(1122):17–40, February 1984.
- [5] R. J. Wootton. Functional morphology of insect wings. *Annu. Rev. Entomol.*, 37:113–140, 1992.
- [6] R. Dudley and C. P. Ellington. Mechanics of Forward Flight in Bumblebees: I. Kinematics and Morphology. *J. Exp. Biol.*, 148(1):19–52, January 1990.
- [7] A. P. Willmott and C. P. Ellington. The mechanics of flight in the hawkmoth *Manduca sexta*. I. Kinematics of hovering and forward flight. *J. Exp. Biol.*, 200(21):2705–2722, November 1997.
- [8] A. P. Willmott and C. P. Ellington. The mechanics of flight in the hawkmoth *Manduca sexta*. II. Aerodynamic consequences of kinematic and morphological variation. *Journal of Experimental Biology*, 200(21):2723–2745, November 1997.
- [9] T. L. Hedrick and T. L. Daniel. Flight control in the hawkmoth *Manduca sexta*: the inverse problem of hovering. *The Journal of experimental biology*, 209(Pt 16):3114–30, August 2006.
- [10] G. J. Berman and Z. J. Wang. Energy-minimizing kinematics in hovering insect flight. *Journal of Fluid Mechanics*, 582:153, June 2007.
- [11] J. P. Whitney. *Design and Performance of Insect-Scale Flapping-Wing Vehicles*. Phd thesis, Harvard University, 2012.
- [12] Q. Wang, J. F. L. Goosen, and F. van Keulen. Optimal hovering kinematics with respect to various flapping-wing shapes. In *IMAV 2013*, Toulouse, France, 2013.
- [13] R. Åke Norberg. The pterostigma of insect wings an inertial regulator of wing pitch. *Journal of Comparative Physiology*, 81(1):9–22, 1972.
- [14] A. J. Bergou, S. Xu, and Z. J. Wang. Passive wing pitch reversal in insect flight. *Journal of Fluid Mechanics*, 591:321–337, October 2007.
- [15] R. Dudley. *The biomechanics of insect flight: form, function, evolution*. Princeton University Press, 2002.
- [16] O. Sotavalta. Flight-tone and wing-stroke frequency of insects and the dynamics of insect flight. *Nature*, pages 1057–1058, 1952.
- [17] C. P. Ellington. The Aerodynamics of Hovering Insect Flight. III. Kinematics. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 305(1122):145–181, February 1984.
- [18] S. P. Sane and M. H. Dickinson. The control of flight force by a flapping wing: lift and drag production. *J. Exp. Biol.*, 204(15):2607–2626, August 2001.
- [19] J. F. L. Goosen, H. J. Peters, Q. Wang, P. Tiso, and F. van Keulen. Resonance Based Flapping Wing Micro Air Vehicle. In *International Micro Air Vehicle Conference and Flight Competition (IMAV2013)*, number September, pages 1–8, Toulouse, France, 2013.
- [20] H. E. Taha, M. R. Hajj, and A. H. Nayfeh. Wing Kinematics Optimization for Hovering Micro Air Vehicles Using Calculus of Variation. *Journal of Aircraft*, 50(2):610–614, March 2013.
- [21] T. Q. Truong, V. H. Phan, S. P. Sane, and H. C. Park. Pitching Moment Generation in an Insect-Mimicking Flapping-Wing System. *Journal of Bionic Engineering*, 11(1):36–51, January 2014.
- [22] H. E. Taha, M. R. Hajj, and P. S. Beran. State-space representation of the unsteady aerodynamics of flapping flight. *Aerospace Science and Technology*, 34:1–11, April 2014.

- [23] S. P. Sane and M. H. Dickinson. The aerodynamic effects of wing rotation and a revised quasi-steady model of flapping flight. *J. Exp. Biol.*, 205(8):1087–1096, April 2002.
- [24] A. Andersen, U. Pesavento, and Z. J. Wang. Unsteady aerodynamics of fluttering and tumbling plates. *Journal of Fluid Mechanics*, 541:65–90, October 2005.
- [25] M. H. Dickinson, F. O. Lehmann, and S. P. Sane. Wing rotation and the aerodynamic basis of insect flight. *Science*, 284(5422):1954–1960, June 1999.
- [26] Y. C. Fung. *An Introduction to the Theory of Aeroelasticity*. Courier Dover Publications, 1993.
- [27] J. P. Whitney and R. J. Wood. Aeromechanics of passive rotation in flapping flight. *Journal of Fluid Mechanics*, 660:197–220, July 2010.
- [28] M. F. M. Osborne. Aerodynamics of flapping flight with application to insects. *Journal of Experimental Biology*, 28:221–245, 1951.
- [29] H. Liu. Integrated modeling of insect flight: From morphology, kinematics to aerodynamics. *Journal of Computational Physics*, 228(2):439–459, February 2009.
- [30] H. J. Peters, J. F. L. Goosen, and F. van Keulen. Flapping wing performance related to wing planform and wing kinematics. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM*, volume 31, pages 1–11, Indianapolis, Indiana, 2012.
- [31] T Weis-Fogh. Energetics of hovering flight in hummingbirds and in *Drosophila*. *Journal of Experimental Biology*, pages 79–104, 1972.
- [32] M. Sun. High-lift generation and power requirements of insect flight. *Fluid Dynamics Research*, 37(1-2):21–39, July 2005.
- [33] NIST/SEMATECH. e-Handbook of Statistical Methods, <http://www.itl.nist.gov/div898/handbook/>.
- [34] DelFly Micro. <http://www.delfly.nl/micro.html>.

Using a MEMS gyroscope to stabilize the attitude of a fly-sized hovering robot

Sawyer B. Fuller[†], E. Farrell Helbling[†], Pakpong Chirattananon, and Robert J. Wood*

ABSTRACT

Creating an autonomous flying vehicle the size of a honeybee presents a number of technical challenges because of its small scale. As vehicle wingspan diminishes, angular acceleration rates increase, necessitating sensing and control systems with high bandwidth. Hovering demonstrations have so far required feedback from high-speed motion capture cameras to estimate the angular velocity, attitude, and position of the vehicle to provide the continuous corrective feedback necessary to avoid tumbling. To achieve autonomous flight, it will be necessary to incorporate a suite of sensors carried onboard. Here we present a step in this direction by integrating a MEMS gyroscope onto an 80 mg flapping-wing micro air vehicle to provide attitude feedback in flight. This enables 2–5 s hovering flights in which the motion capture system provides only position feedback. Our vehicle, and likely others in the future of similar scale, is propelled by flapping wings that generate vibration. Our results indicate that the resulting accelerations, measured as high as 5 g by the sensor's accelerometer, significantly distort readings from the accelerometer but not the gyroscope.

1 INTRODUCTION

Significant challenges accompany the reduction of a flying robot to the size of a fly or bee. As vehicle size diminishes, conventional propulsion approaches (e.g. as in rotorcraft) become impractical because of physical scaling laws. For example, energy losses per unit mass of an electromagnetic motor increase as l^{-2} [1], where l is some characteristic length of the vehicle such as the chord length. Biological organisms at this scale instead use flapping wings [2]. Most small flying animals, from hummingbirds to flies, have evolved a convergent solution of hovering by flapping wings to overcome the low glide ratio of fixed wings at low Reynold's numbers [3]. The resulting unsteady aerodynamics augment lift [2, 4].

*[†]Authors contributed equally. All authors are with the School of Engineering and Applied Sciences and the Wyss Institute for Biologically Inspired Engineering, Harvard University, Cambridge MA 02138 USA (E-mail: minster@seas.harvard.edu, ehelbling@seas.harvard.edu, pakpong@seas.harvard.edu, and rjwood@eecs.harvard.edu)

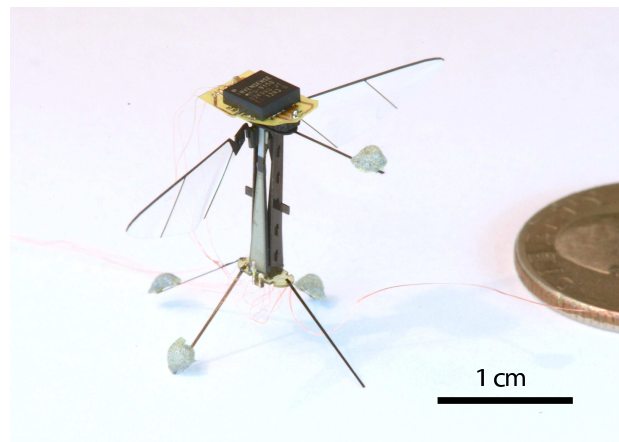


Figure 1: A robotic fly (81 mg) carries an onboard inertial measurement unit (40 mg) attached to the anterior to measure angular velocity. This feedback was used by the controller to apply corrective torques through the wings to stabilize its flight and hover.

The challenges at small scale extend to sensing and control. As the vehicle becomes smaller, the rate of rotational acceleration increases, scaling as l^{-1} [5]. This challenge is exacerbated by dynamic instability for designs that have the same basic body plan as hovering insects such as flies and bees, whose body hangs below the wings. Aeromechanics simulations suggest that typical baseline hovering wing kinematics lead to a dynamically unstable system [6, 7, 8]. Therefore, not only must a flight controller for such vehicles perform continuous corrective maneuvers as is required in fighter jets [9], it must do so with a time delay that is orders of magnitude shorter because of the smaller length scale. This is because fundamental control limits place an upper bound on the time delay that can be tolerated in a control loop that must stabilize a dynamically unstable system [10]. A vehicle the size of a fly or bee is nevertheless large enough that it is foreseeable that currently-available sensing and computation technologies could be miniaturized to fit within its mass and power constraints. An initial step toward flight autonomy is to equip a sensor that can stabilize its attitude dynamics.

The first fly-sized robot to carry its own weight overcame the propulsion challenge by driving flapping wings using piezoelectric actuators [4, 11]. Piezoelectric actuators have no moving components, allowing them to scale downward more favorably compared to motors that require

bearings and gears [1]. A complementary laser-based microfabrication process was developed to quickly and precisely create flexure-based articulated mechanisms out of high-performance materials such as carbon fiber and polyimide [12].

Here we consider the attitude control problem of such a vehicle. Advances in fabrication and design led to an 81 mg prototype with the actuation capability to control its flight motions [13], shown in Figure 1. Using an array of external motion capture cameras and reflective markers attached to the vehicle to provide attitude and position feedback, a controller was developed that enabled it to perform controlled flight maneuvers [14]. To fly this vehicle autonomously outside of laboratory conditions, however, scale-appropriate sensing and control must be developed. Stability has previously been achieved with airfoils such as air dampers [7] or a tail [15, 16], but this makes the vehicle susceptible to wind disturbances and sacrifices maneuverability. An alternate approach, inspired by the ocelli of flying insects, achieved upright stability using four light-sensitive phototransistors and a light source above the vehicle [17]. In this work we use an inertial measurement unit (IMU) built from micro-electromechanical systems (MEMS) to provide information about the attitude of the vehicle so that it can stabilize its flight motions. Feedback from onboard gyroscopes has previously been used on flapping-wing vehicles, such as a 20 g hummingbird vehicle [18] and a 13 g four-winged ornithopter [15], but these vehicles are more than two orders of magnitude more massive.

Key questions to be addressed in the current study are whether the vehicle can be tuned to stably carry the weight of the sensor, which adds approximately 50% to the total vehicle weight, whether data can be returned through the tether with high fidelity, and whether the sensor's operation is disrupted by the high-frequency vibratory environment of flapping flight. In [18], it was found that certain MEMS gyroscopes produce nonsensical output when mounted on a flapping-wing vehicle. Gyroscope output was numerically integrated successfully to estimate absolute attitude for short periods on Baek's ornithopter [15]. Both of these platforms, however, flap at only ~20 Hz. It is therefore unknown whether sensor output will be disrupted at the much higher 120 Hz flapping frequency of our vehicle.

Our results show that by carefully calibrating the vehicle, it can carry the extra payload of the sensor. We find that neither communication nor gyroscopic fidelity is significantly disrupted by deployment on the tiny flapping-wing vehicle. In Sections 2 and 3, we describe the MEMS IMU and the flapping-wing micro air vehicle used in our test flights. In Section 4 we give an analysis of how rate feedback can give upright stability and demonstrate this principle using flight tests, reproducing earlier results using ocelli [17]. In Section 5 we show how this sensor was used to achieve short duration hovering, by demonstrating a series of flight tests

Table 1: Characteristics of the MPU9150 Inertial Measurement Unit

Characteristic	Typical Value
Mass	36 mg
Area	$4 \times 4 \text{ mm}^2$
Height	1 mm
Power	2.4 V, 9.4 mW
Max. Output Data Rate	
Gyroscope	8 kHz
Accelerometer	1 kHz
Magnetometer	8 Hz
Max. Range	
Gyroscope	$\pm 2000^\circ/\text{s}$
Accelerometer	$\pm 16g$
Magnetometer	$\pm 1200 \mu\text{T}$
Max. Sensitivity	
Gyroscope	16.4 LSB/($^\circ/\text{s}$)
Accelerometer	2048 LSB/g
Magnetometer	0.3 LSB/ μT

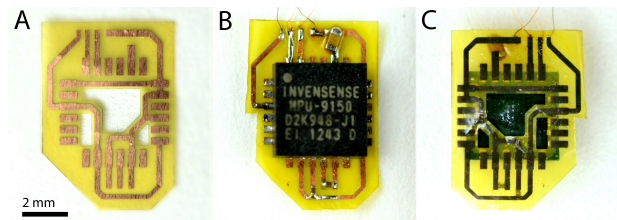


Figure 2: IMU connected to the custom flex circuit. A) Unpopulated circuit board. B) Sensor and three capacitors (two 01-005 and one 04-02 size) after attachment to the circuit board, showing four-wire connection at the top. The outer border of the flex circuit has been modified from A to B to accommodate placement on the top of the robot. C) Populated circuit board shown from the back.

in which successively fewer components of external motion capture measurements are used for feedback control.

2 MEMS INERTIAL MEASUREMENT UNIT

2.1 Selection and Fabrication

The MPU9150 (Invensense, San Jose, CA, U.S.A.) was the first MEMS device to include a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer in a package small enough for our robotic fly to carry (see Table 1 for specifications). It communicates using the I2C protocol, providing feedback rates as high as 1 kHz. This is a higher rate than the 500 Hz motion capture system which has previously been used to stabilize the robot [14]. The sensor requires a minimum of three capacitors—one bypass capacitor, a filter capacitor and a charge pump capacitor. In addition, the pins for external clock input, frame synchronization input, and the

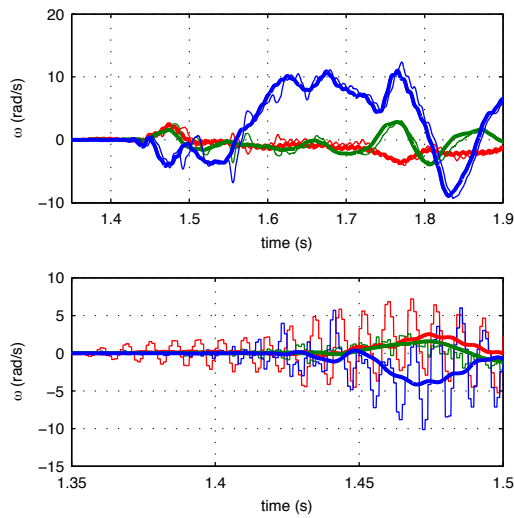


Figure 3: (top) Data from the flight in Figure 6 comparing filtered estimates of angular velocity ω from the gyroscope (thick lines) to that of the motion capture system (thin lines). The wings ramp-up flapping starting at $t=1.35$ s; red, green, and blue correspond to ω_x , ω_y , and ω_z , respectively. Spikes in the motion tracking estimates from the period $t=1.5$ to 1.7 s are due to imperfect tracking. (bottom) A close-up of readings from the gyroscope before filtering (thin lines, 1 kHz sampling frequency can be observed) and after filtering by a 3rd order low-pass Butterworth filter with a 40 Hz cutoff frequency. A prominent oscillation at the 120 Hz flapping frequency of the wings can be observed in the unfiltered measurements.

slave address least significant bit need to be pulled low. The logic voltage and supply voltage were connected to reduce the number of wires and weight. To satisfy these requirements we designed a custom printed circuit board (flex circuit) to mount on the robot.

The sensor itself has a mass 36 mg (Table 1). The additional components of the capacitors, board, and solder added an additional 4 mg. In [13], measured lift forces indicated that the robotic fly could carry an additional 60-70 mg while flapping with a maximum wingbeat amplitude. But operating the wings at full amplitude leaves no capability to modulate wing kinematics to apply torques around the pitch and roll axes (Section 3). The lower the payload, the greater the control torques that can be applied. Accordingly, to minimize mass the board was made from thin copper-clad flex circuit material (18 μm copper, 12.7 μm polyimide). The circuit was fabricated using a direct-write photolithography and etching process and components were then hand-soldered (Figure 2). To minimize torque due to an off-axis placement, the sensor was attached at the anterior of the vehicle, as shown in Figure 1.

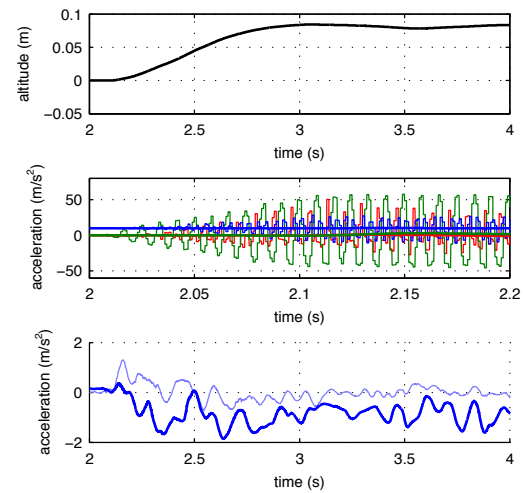


Figure 4: Accelerometer measurements in flight. (top) In a hovering test, the vehicle begins flapping its wings at $t=2$ s and flies to the setpoint altitude. (middle) A plot of accelerometer readings over a shorter time window shows the 120 Hz oscillations induced by the flapping wings and filtered version (approximately horizontal lines). Red, green, and blue are body x -, y -, and z -axis measurements, respectively. Accelerations up to ± 5 g are measured along the forward-backward direction of the flapping wings. (bottom) Motion capture estimate of acceleration (filtered, light line) compared to accelerometer measurements along the z -axis (filtered, dark line, data rotated to world frame).

2.2 Communication interface and control

Sensor data was communicated over I2C at 400 kbps through four 51-gauge copper wires measuring approximately 30 cm. Control computations were performed by a desktop computer running an xPC Target realtime operating system (Mathworks, Natick, MA, USA). The data rate is sufficient to communicate gyroscope and accelerometer readings at a rate of 1 kHz. The control loop on this computer operated at 10 kHz to provide smooth analog signals to external high-voltage amplifiers. Because the I2C protocol requires digital communication at rates much higher than the control loop frequency, a field-programmable-gate-array (FPGA) add-in card (Speedgoat GmbH, Liebfeld, Switzerland) was programmed to implement the I2C protocol. A voltage level shifter (Phillips GTL2002) allowed digital electrical communication between the MPU9150 (max 3.3 V) and the FPGA board (5V TTL). Embedded Matlab software translated the 16-bit signed integer sensor readouts into floating point quantities for use in the controller.

2.3 Initial Characterization

To characterize the sensor, we compared the outputs of the gyroscope and the accelerometer to motion capture measurements during open and closed-loop flight experiments

(see Figure 3 (top) for comparison). The filtered gyroscope data reproduces the motion capture measurement faithfully, with approximately 10 ms less latency.

As can be seen in Figure 4, the measured acceleration along the z -axis of the robot can generally track the motion capture estimate; however, when the wings begin flapping at approximately $t=2.1$ s, the accelerometer estimate becomes negative (Figure 4). Given the altitude trajectory, we expect a positive acceleration at the start that approaches zero as the robot reaches the desired altitude. This is reflected in the motion capture measurement. The raw signal from the accelerometer, however, shows high frequency noise with a maximum amplitude of (± 5 g) along the y -axis. This is in the same direction as the forward-backward motion induced by the flapping wings. It is possible this large oscillation cross-couples to other degrees of freedom or moves the sensor into a nonlinear regime in which it is slightly less precise. The distortion is nearly -2 m/s², is larger than the accelerations of interest, which are on the order of 1 m/s² (Figure 4). Integrating this sensor's output twice to estimate position gives an error on the order of one meter within a second in all directions, suggesting it cannot be used for motion estimation except at very short timescales.

3 ROBOTIC FLY

As described in more detail in [13, 14], the robotic fly prototype considered here (Figure 1) has a pair of independently-actuated wings that can produce sufficient lift to take off and induce torques independently about orthogonal axes to control the vehicle's motion. We define a right-handed coordinate system for the body in which the right wing points along the positive x -axis, the y -axis points forward, and the z -axis points upward (Figure 5). In this definition, the long axis of the body hangs downward in the negative- z direction. Roll torque is induced by varying the relative stroke amplitudes of the left vs. right wing. Pitch torque is induced by moving the "mean stroke angle"—the time-averaged angle of the forward-backward motion of the wings—in front ($+y$) or behind ($-y$) the center of mass [13].

The robot is given power and controlled through a lightweight compliant tether wire that has little effect on vehicle dynamics [14, 17]. Power and control commands are transmitted over four 51-gauge (0.022 mm diameter) copper wires and feedback from the IMU is returned through a second bundle of four. Limitations in the current manufacturing process limit vehicle lifetime of each laboriously-fabricated vehicle [11] to just a few minutes of flight, after which flexure joints fail and the vehicle must be replaced. Accordingly, only a few flights were possible in this study.

4 UPRIGHT STABILITY IN FLIGHT

The simplest way to utilize a gyroscope to stabilize flight motions is to take advantage of the natural dynamics of the vehicle. In previous work [17] it was shown analytically and

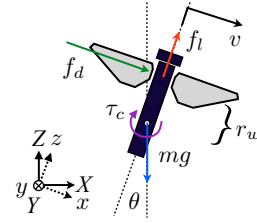


Figure 5: Model and axis convention for the robotic fly. The vehicle state is given by three quantities, the rotation angle θ , the rotation velocity ω , and the lateral velocity in body coordinates v . A lift force f_l generated by the flapping wings acts at a distance r_w away from the center of mass and along the body- z direction, an aerodynamic drag force f_d acts at a distance r_w from the CM, and the gravitational force mg acts at the center of mass. The right-handed axis convention for world coordinates is shown, with y pointing into the paper. A feedback controller applies a control torque τ_c by altering baseline wing kinematics.

experimentally on the flapping-wing robot testbed that stability of the vehicle can be achieved by applying a torque proportional to the angular velocity in the pitch and roll directions. This is because the body of the robotic fly hangs below the wings, exhibiting pendulum-like dynamics that are unstable without feedback, but can be stabilized through a velocity-proportional feedback that induces a "damping" effect. In that work, an ocelli-inspired light sensor was used to estimate angular velocity. Here, we first aimed to verify whether angular velocity feedback measured from the gyroscope could be used as a substitute.

4.1 Stability Analysis

A diagram of the planar dynamics is shown in Figure 5. A test of this vehicle flapping in a wind tunnel indicated that the stroke-averaged drag force on the wings is nearly linear with the incident airspeed for typical wing kinematics. This is the case for wind in both the x - and y -directions. Accordingly, the model for aerodynamic drag in both cases is:

$$f_d = -b_w v_w,$$

where v_w is the lateral velocity of the point on the airframe at the midpoint between the two wings and b_w is a drag constant. If the vehicle is rotating at angular velocity ω , then the velocity of the wings, when linearized around $\theta = 0$, is $v_w = r_w \omega + v$, where r_w is the distance from the midpoint of the wings to the CM. Similarly, the force arising from aerodynamic drag is $f_d = -b_w(v + r_w \omega)$ and the torque about the center of mass due to this force is $\tau_d = -r_w f_d = b_w r_w v - b_w r_w^2 \omega$. With lift force due to the flapping wings (f_l) approximately balancing out weight (mg), the lateral force due to the tilted thrust vector is equal to $-f_l \sin \theta = -mg \sin \theta \approx -mg \theta$ for small θ .

With a control law that applies a torque proportional to the rate of rotation,

$$\tau_c = -k_d \omega, \quad (1)$$

and assuming planar motion, we can neglect second-order cross-product terms and equate forces and torques to velocities according to $f = m\dot{v}_x$ and $\tau = J\dot{\omega}$. The linearized equations of motion about zero pitch angle θ can be written as a state-space dynamical system $\dot{\mathbf{q}} = \mathbf{A}\mathbf{q}$ with the state vector $\mathbf{q} = [\theta, \omega, v]^T$ expressed in body coordinates, where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{1}{J}(b_w r_w^2 + k_d) & \frac{1}{J}b_w r_w \\ -g & \frac{b_w r_w}{m} & -\frac{b_w}{m} \end{bmatrix}. \quad (2)$$

Using the Routh-Hurwitz stability criterion gives a lower bound on the velocity feedback gain in the controller given in Equation (1) necessary to achieve asymptotic stability at $[\theta, \omega, v] = \mathbf{0}$ for the robotic fly. Using previously-published parameters and the addition of the 40 mg mass of sensor on the anterior of the robot, 1.5 mm above the leading edge of the wing, we found that a gain of $k_d = 2 \times 10^{-7}$ was sufficient to achieve stability for both xz - and yz -dynamics.

4.2 Flight Tests

Tuning in preparation for closed-loop flights was initiated as follows. If the vehicle is sufficiently well-fabricated that it can take off without a torque bias, then the direction of pitch or roll velocity after take-off is observed. Trim values for pitch and roll torque are altered to the maximum extreme to oppose its initial torque. If in the subsequent take-off the vehicle is observed to apply a net torque in the opposite direction, the vehicle is deemed worthy of further testing. A search is then performed to find trim parameters for which takeoff is nearly vertical.

The vehicle's flight trajectory with feedback from the gyroscope alone is shown in Figure 6. In this flight, feedback from the gyroscope was applied according to the control law in Equation (1). The vehicle remains stably upright during this flight, indicating that gyroscopic feedback is able to stabilize its upright orientation as desired. Without feedback, in all trials yet observed, the vehicle quickly tumbles.

5 HOVERING

In this section we build on the demonstration of achieving upright stability using gyroscope feedback to demonstrate hovering using a combination of gyroscope feedback and motion capture.

The hovering controller used here incorporates improvements over that reported in [14] that includes an adaptive component that continuously estimates vehicle parameters such as torque offsets [19]. We provide an overview here. The flight controller requires three inputs: angular velocity ω (body coordinates), attitude $[\hat{x}, \hat{y}, \hat{z}] = R^T$ (the unit vectors of the axes of the vehicle in world coordinates, equivalent to the transpose of the rotation matrix R), and position

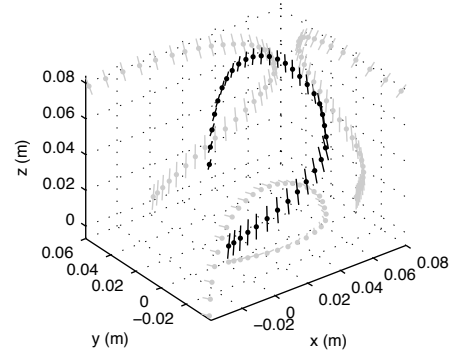


Figure 6: Plot of flight trajectory in which angular velocity feedback from the gyroscope was used to maintain an upright orientation. In this flight, position was not under feedback control. Vehicle attitude is rendered at 20 ms intervals. Dots represent the location of the CM of the vehicle and the black line denotes the long axis of the body; projections of the trajectory are shown in grey in the background. In the z -projection below the vehicle, the “forward” $+y$ direction of the vehicle is projected rather than its long axis to indicate the vehicle's heading.

$[X, Y, Z]^T$ (world coordinates). An independent altitude controller modulates the thrust force to maintain the desired altitude given position feedback. An outer position loop computes a desired direction of the $\hat{z}$ vector, $\hat{z}_d$, based on the position error. By inclining the axis of the vehicle, lateral thrusts can be generated. The attitude error is defined as $\mathbf{e} = [\hat{y} \cdot \hat{z}_d, -\hat{x} \cdot \hat{z}_d, 0]^T$. Based on this error, the attitude controller calculates torques as a function of the attitude error $\mathbf{e}$ and angular velocity ω according to:

$$\begin{aligned} \tau_c &= f(\mathbf{e}, \omega) + \tau_o \\ &= -K_a(\omega + \Lambda \mathbf{e}) - (\Lambda \mathbf{e} \times \mathbf{J}\omega) - \mathbf{J}\Lambda \dot{\mathbf{e}} + \tau_o \end{aligned} \quad (3)$$

where K_a and Λ are positive diagonal gain matrices that are experimentally tuned, τ_o is the trimmed offset torque necessary to compensate for manufacturing irregularity, and $\mathbf{J}$ is the estimate of the inertia matrix.

To use gyroscope feedback in this controller, note that the output of the three-axis gyroscope is an estimate of ω . Additionally, from Equation (3), an estimate of attitude error $\mathbf{e}$ is needed. We parameterize the vehicle attitude using a zyx -convention for Euler Angles to avoid representation singularities at hover. In this convention, attitude is defined by first rotating the frame by an angle $\theta_3 \in (-\pi, \pi]$ (yaw) around the body z -axis, then $\theta_2 \in (-\frac{\pi}{2}, \frac{\pi}{2})$ (roll) around the new body y -axis, and then $\theta_1 \in (-\frac{\pi}{2}, \frac{\pi}{2})$ (pitch) around that body x -axis. Performing the same rotations about world axes but in the opposite order, that is, XYZ , gives the same attitude. With this convention, the rates of change of these angles as a function of the angular velocity vector ω are given by the

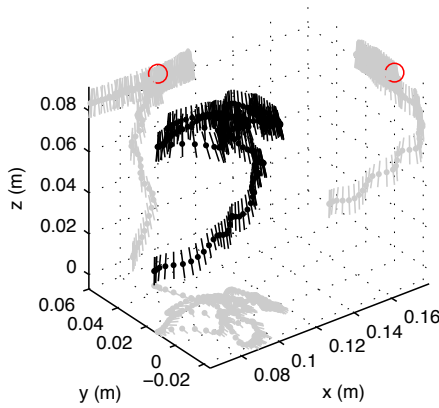


Figure 7: Plot of flight trajectory, rendered at 20 ms intervals. This flight used feedback entirely from motion capture, providing a baseline performance measure. The desired hovering position is indicated by a red circle. Other figure conventions are the same as in Figure 6.

product

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} \cos \theta_3 / \cos \theta_2 & -\sin \theta_3 / \cos \theta_2 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 \\ -\cos \theta_3 \tan \theta_2 & \sin \theta_3 \tan \theta_2 & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \quad (4)$$

By integrating Equation (4) over time, an estimate of the attitude can be computed from the rotation matrix derived from these Euler Angles [19], from which e and its derivative can be computed.

Because of the limited lifetime of the robot, we performed the experiments incrementally. First, we verified that hovering was operational using only feedback from motion capture, insuring that the necessary amount of control authority was available after incorporating the additional payload, and to quantify a baseline flight performance. Next, we incorporated the angular velocity estimate ω from the gyroscope into the feedback loop. Finally we additionally incorporated the attitude estimate $\hat{z}$ derived from integrating gyroscope output (Equation (4)) so that motion capture feedback was entirely eliminated from attitude control of the robot.

5.1 Motion Capture Feedback

Figure 7 shows that, with appropriate tuning of controller gains and trim values, the robot was able to carry the extra payload and take off and hover near the desired setpoint position $[X, Y, Z]^T = [0.1, 0, 0.06]^T$ m using motion capture feedback. Its mean position for the 2 s period after reaching the desired altitude was $[0.109, 0.0077, 0.056]^T$ m.

5.2 Angular Velocity Feedback

Having demonstrated in Section 2 that the angular velocity estimate of ω from the gyroscope accurately tracks the ground truth measurement from motion capture, with lower

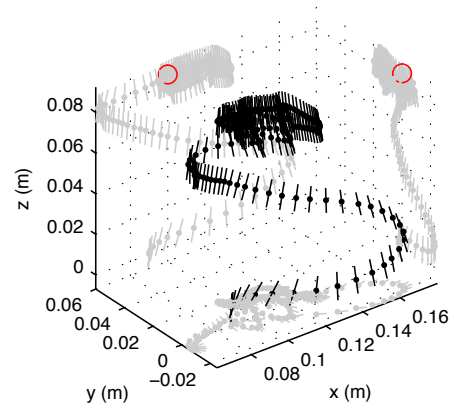


Figure 8: Plot of hovering flight trajectory in which gyroscope provided an estimate of the angular velocity ω . Other details are the same as in Figure (7).

latency, we incorporated this estimate into the hovering attitude controller. Figure 8 shows that the robot achieves hovering flight, flying near the set point, maintaining a mean position of $[0.093, 0.0058, 0.056]^T$ m and comparable error.

5.3 Attitude Feedback

In this experiment, we performed a zero-order hold numerical integration of Equation (4), initialized using the motion capture estimate of attitude at $t=0$ s. Figure 9 shows the trajectory of the robot during a flight in which both angular velocity and attitude were estimated by feedback from the onboard gyroscope. In this experiment the robot remained near the setpoint position for approximately 2 s, after which it began to exhibit significant position error. This may be attributable to drift in the attitude estimate causing a lateral thrust error. During the initial 2 s period of tracking, the robot maintained a mean position of $[0.075, 0.029, 0.057]^T$ m.

5.4 Analysis

The flight using only motion capture estimates produced a mean Euclidean distance error of less than 0.01 m from the setpoint. When the ω estimate from the gyroscope was incorporated, the mean error was a comparable 0.01 m. For the flight in which gyroscope feedback additionally estimated the attitude $\hat{z}$, the error increased to 0.03 m. A possible explanation for the error is that, as can be observed in Figure 10, a visible drift is noticeable along the pitch (red) axis, reaching an error of 6° (0.1 rad). This drift, perhaps arising because the largest oscillations occur about this axis because of the forward-backward motions of the wings (Figure 3), would cause a force error in the y -direction as the vehicle tilted in response.

In addition to plotting the 3D trajectory of each flight and comparing the mean position of each flight experiment, we calculated the root mean square error (RMSE, see Figure 11). We expect comparable levels of RMSE in altitude because

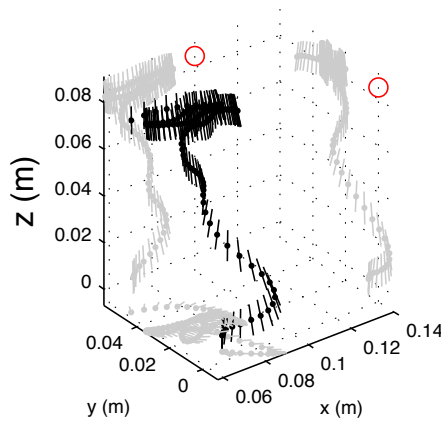


Figure 9: Plot of flight trajectory in which the gyroscope provided all attitude feedback in the flight controller, including both angular velocity ω and attitude $\hat{z}$. In this flight, the position error was somewhat larger than in Figure (8), possibly because of drift in the attitude estimate. Other details are the same as in Figure (7).

the altitude controller runs independently of the attitude controller and receives the same measurements in all tests, as can be observed in Figure 11. RMSE is comparable between the motion capture-only experiment and experiment in which the gyroscope estimated ω for both x -position, where the errors are 1.17 cm and 1.39 cm, respectively, as well as for the y -position, where the errors are 1.26 cm and 1.05 cm, respectively. The RMSE of the lateral position of the flight with attitude estimation from the gyroscope is significantly larger at 2.72 cm in x -position and 2.95 cm in y -position. Given that the robot had difficulty correcting its pitch angle and drifted away from the set point during flight, these values are not unexpected. These results indicate that estimates from the gyroscope can be substituted for those from the motion capture system and still maintain stability and control of the vehicle despite its fast and unstable dynamics.

6 CONCLUSIONS

In this work we demonstrated a MEMS IMU integrated with a fly-sized robotic vehicle. Feedback from this sensor was shown to be sufficient to stabilize this vehicle's upright orientation and estimate the vehicle's attitude for short periods for use in an inner attitude control loop. This constitutes the first demonstration of controlled flight in fly-sized robot in which only position feedback was derived from external sensors.

Our results indicate that gyroscope sensor readings are somewhat, though not significantly impaired by vibrations induced by the wings. Because the flapping frequency is known, a future improvement could refine the filter design to better eliminate it, either by a notch filter or by adaptive cancellation, and thus reduce feedback latency even further.

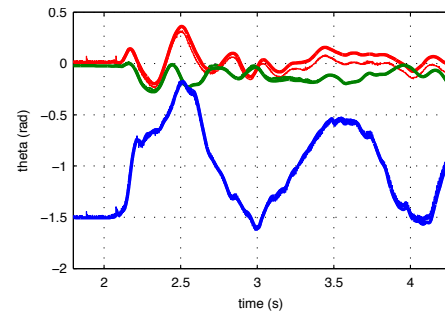


Figure 10: Comparison of measured Euler Angles – pitch (red), roll (green), and yaw (blue) – from motion capture (thin line) to the estimate derived by integrating gyroscope feedback (thick line) from the flight shown in Figure (9). The flight starts at $t=2.0$ s. Drifting over time can be observed, particularly about the x -axis (pitch).

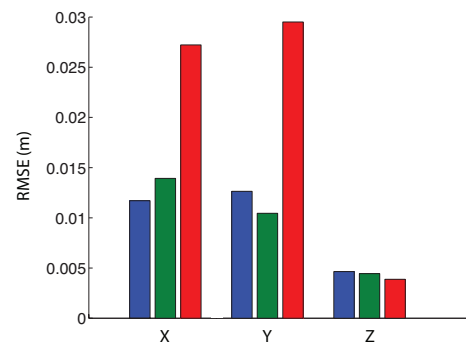


Figure 11: Root Mean Square Error (RMSE) position error for x -, y -, and z -coordinates for three flight experiments: motion capture feedback (blue), ω estimate from gyroscope (green), and full attitude measurements ($\omega, \hat{z}$) from gyroscope (red).

Our results indicate that the accelerometer component exhibited significant noise and distortion in the presence of the flapping wings. This suggests alternative sensing modes may be necessary to quickly sense translational motions for high-bandwidth control, such as infrared proximity sensors, vision, or a pressure sensor (for altitude). We cannot rule out that sonar or laser rangefinders could be sufficiently miniaturized.

As the fabrication process matures, more flights will be possible, which will permit a more comprehensive characterization of the system dynamics presented here. To permit longer flight times, drift in the attitude estimate could be reduced by incorporating accelerometer or magnetometer feedback using a Kalman Filter.

The results reported here provide a baseline sensor suite for stable flight that will enable future flight control studies. They motivate further development in lightweight sensing,

computation, power source, and control technologies appropriate for bee- or fly-sized flying vehicles.

7 ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation (award numbers CMMI-0746638 and CCF-0926148) and the Wyss Institute for Biologically Inspired Research. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

- [1] W. S. N. Trimmer. Microbots and micromechanical systems. *Sensors and Actuators*, 19:267–287, 1989.
- [2] Robert Dudley. *The biomechanics of insect flight: form, function, evolution*. Princeton University Press, 2002.
- [3] U. Pesavento and Z.J. Wang. Flapping wing flight can save aerodynamic power compared to steady flight. *Physical review letters*, 103(11):118102, 2009.
- [4] R. J. Wood. The first takeoff of a biologically inspired at-scale robotic insect. *IEEE Transactions on Robotics*, 24(2):341–347, 2008.
- [5] Vijay Kumar and Nathan Michael. Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research*, 31(11):1279–1291, 2012.
- [6] Leif Ristroph, Gunnar Ristroph, Svetlana Morozova, Attila J. Bergou, Song Chang, John Guckenheimer, Z. Jane Wang, and Itai Cohen. Active and passive stabilization of body pitch in insect flight. *Journal of The Royal Society Interface*, 10(85), 2013.
- [7] Zhi Ern Teoh, Sawyer B. Fuller, Pakpong C. Chirattananon, Nestor O. Perez-Arancibia, Jack D. Greenberg, and Robert J. Wood. A hovering flapping-wing microrobot with altitude control and passive upright stability. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3209–3216, Vilamoura, Algarve, Portugal, 7–12 October 2012. IEEE.
- [8] Imraan Faruque and J. Sean Humbert. Dipteran insect flight dynamics. Part 1: Longitudinal motion about hover. *Journal of Theoretical Biology*, 264(2):538–552, 2010.
- [9] Malcolm J. Abzug and Eugene E. Larrabee. *Airplane stability and control, Second Edition: A history of technologies that made aviation possible*. Cambridge University Press, Cambridge, MA, 2002.
- [10] Karl Johan Astrom and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 41 William Street, Princeton, NJ 08540, 2008.
- [11] Robert J. Wood, Ben Finio, Michael Karpelson, Kevin Ma, Nestor O. Pérez-Arancibia, Pratheev S. Sreetharan, Hiroto Tanaka, and John P. Whitney. Progress on “pico” air vehicles. *The International Journal of Robotics Research*, 31(11):1292–1302, 2012.
- [12] JP Whitney, PS Sreetharan, KY Ma, and RJ Wood. Pop-up book mems. *Journal of Micromechanics and Microengineering*, 21(11), 2011.
- [13] Kevin Y. Ma, Samuel M. Felton, and Robert J. Wood. Design, fabrication, and modeling of the split actuator microrobotic bee. In *Proceedings of the IEEE Int Robotics and Automation Conference*, pages 1133–1140, St. Paul, MN, 14–18 May 2012. IEEE.
- [14] Kevin Y. Ma, Pakpong Chirattananon, Sawyer B. Fuller, and Robert Wood. Controlled flight of a biologically inspired, insect-scale robot. *Science*, 340(6132):603–607, 2013.
- [15] Stanley S Baek, Fernando L Garcia Bermudez, and Ronald S Fearing. Flight control for target seeking by 13 gram ornithopter. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2674–2681. IEEE, 2011.
- [16] C. De Wagter, S. Tijmons, B.D.W. Remes, and G.C.H.E. de Croon. Autonomous flight of a 20-gram flapping wing mav with a 4-gram onboard stereo vision system. In *Robotics and Autonomous and Systems (ICRA), 2014 IEEE/RSJ International Conference on*, Hong Kong, China, 2–5 June 2014. IEEE.
- [17] Sawyer B. Fuller, Michael Karpelson, Andrea Censi, Kevin Y. Ma, and Robert J. Wood. Controlling free flight of a robotic fly using an onboard vision sensor inspired by insect ocelli. *Journal of The Royal Society Interface*, 11(97), August 2014.
- [18] Matthew Keennon, Karl Klingebiel, Henry Won, and Alexander Andriukov. Development of the nano hummingbird: A tailless flapping wing micro air vehicle. In *AIAA Aerospace Sciences Meeting*, pages 1–24, Reston, VA, 9–12 January 2012. AIAA.
- [19] Pakpong Chirattananon, Kevin Y Ma, and Robert J Wood. Adaptive control of a millimeter-scale flapping-wing robot. *Bioinspiration & biomimetics*, 9(2):025004, 2014.

Effect of chordwise deformation on unsteady aerodynamic mechanisms in hovering flapping flight

T. A. Noyon^{1,*}, W. B. Tay^{1,2}, B. W. van Oudheusden¹, and H. Bijl¹

1. Delft University of Technology, Delft, The Netherlands

2. Temasek Laboratories, National University of Singapore, Singapore

ABSTRACT

A three-dimensional simulation of hovering flapping wings was performed using an immersed boundary method. This was done to investigate the effects of chordwise wing deformation on three important unsteady aerodynamic mechanisms found in flapping flight, namely Leading Edge Vortex (LEV) shedding, wake capture and clap and fling. A wing was modeled as a flat plate, flapping close to a symmetry plane. Three different deforming chords were defined, a rigid case, a case with maximum deformation at the trailing edge and increased angle of attack (AoA) near the leading edge, and a case with the maximum deformation in the center of the chord and decreased AoA near the leading edge. All cases had zero deformation at the wing root and maximal deformation at the wing tip. A higher AoA near the leading edge resulted in faster LEV buildup and faster buildup of lift. No shedding of the LEV was observed in any of the cases even when deformation caused a high AoA near the leading edge. A distinct dip in lift buildup was observed and shown to be caused by the interaction between the previously shed vortex and the newly developing LEV. This interaction occurred faster when the AoA at the leading edge was increased, and slower when the angle of attack was decreased. Moving the wing closer to the symmetry plane had a positive effect on the cycle average value of C_L . This positive effect was reduced however by the earlier interaction between the LEV and the previously shed vortex.

1 INTRODUCTION

In view of the Reynolds number range at which MAVs operate and the desired flight performance flapping flight concepts are considered beneficial, since fixed-wing aircraft do not have the desired maneuverability and helicopters are too inefficient and noisy [1]. This has motivated designers to look to nature for inspiration, since insects are extremely maneuverable, silent and can be more efficient at low flight speeds

[2]. The aerodynamics of insect flight are very different from those of fixed wing aircraft or rotorcraft however, since it exploits unsteady aerodynamic mechanisms to generate sufficient lift force, including a stable leading-edge vortex (LEV), clap and fling use of rotational mechanisms and wake capture [3]. Biologists have been interested in the flying capabilities of insects for a long time and have conducted many experiments using free flying insects [3], tethered insects [4] and dynamically scaled mechanical flappers [4, 5, 6]. To further understand the effect of all relevant parameters it is essential to quantify and visualize the three-dimensional flow around the wings. Although recent developments in non-intrusive measurement techniques allow researchers to capture three dimensional flow fields [7, 8], it remains very difficult to capture all relevant details of the flow using only experimental techniques. Therefore numerical simulations can provide additional information for understanding the flow phenomena [9]. Several three-dimensional simulations have been performed on specific insect geometries, however, the computational requirements of such simulations are too demanding to do a systematic parametric study of all the important parameters involved. More general studies on the wing flexibility have used two-dimensional simulations and have particularly noted the positive effect of a more flexible wing on leading edge vortex (LEV) stability [10, 11, 12], which in turn has a large positive effect on the lift force. The LEV stability is also affected by three dimensional flow however [13, 14], so the benefit of a flexible wing in a three dimensional case cannot be determined from a two-dimensional simulation alone. Studies investigating the three-dimensional effects in flapping flight using simulations with rigid wings have shown that the tip vortex created at the free end of the wing stabilizes the LEV, from which can be concluded that this stability is also affected by the aspect ratio of the wing [15] and the kinematics of the wing [16]. To get a complete picture of how different parameters affect the LEV stability, and with that the force coefficients of the wing, a three-dimensional simulation is required.

The objective of the present study is to perform a three-dimensional simulation of hovering flapping wings, in order to investigate the effects of chordwise wing deformation on three important unsteady aerodynamic mechanisms found in flapping flight, namely LEV shedding, wake capture and clap and fling. This will result in a better understanding of the physics involved in flapping flight, which in turn could lead to

*Email address(es): tijsnoyon@gmail.com

improved MAV designs. Since the actual three-dimensional deformation of both insect and MAV wings is very specific to the wings' structural and material properties, this study does not attempt to include fluid-structure interaction, instead prescribing the time dependent deformation of the wings directly. The direct prescription of the wing deformation allows a more controlled way to investigate its effect on the unsteady aerodynamic mechanisms. The configuration of the wings that will be simulated is based on a wing pair as found on a four-winged MAV in biplane configuration such as the DelFly [17]. This means the wings will rotate in opposing phase around a point at the leading edge root. Under this condition the flow around one wing will be affected by the presence of the other wing. The wing interaction effects will be modeled by creating a symmetry plane, and investigated by varying the distances to the symmetry plane. Only deformation in chordwise direction will be investigated in this study, with the wing considered fully rigid in spanwise direction. However, the chordwise deformation will be varied along the span, resulting in a twist in spanwise direction.

2 NUMERICAL METHOD

A serious issue in simulating flapping wings is the large translational and rotational motions of the wings. In conventional body conforming grid simulation methods these large movements result in large grid deformations, which causes problems in convergence and stability of the algorithms as well as having a negative impact on the accuracy [18]. The Immersed Boundary Method (IBM) used in this study is a combination of the methods described by [19], [20] and [21]. In both the method by [21] and the one by [20] the forcing term fc is calculated explicitly using an Adam-Bashforth second-order (AB2) scheme. In this study, fc is calculated explicitly using the first-order forward Euler and second-order AB2 schemes for the viscous and convective terms, respectively. This will reduce the computational cost while having no observable difference on the results compared to an AB2 scheme [22]. The forcing term fc will be calculated according to Equation 1.

$$fc^{n+1} = \frac{u_f - u^n}{\Delta t} + \left[u^n \cdot \nabla u^n - \frac{1}{Re} \nabla^2 u^n \right] + \nabla p^n \quad (1)$$

In this equation the superscript indicates the time step number and u_f is the velocity contribution of the boundary which has to be imposed. Since the boundary does not match the grid this velocity is not known and is obtained through a simple triangle linear interpolation of the interface and three points in the surrounding velocity field [22]. Finally the total force exerted on the surface of the solid will be calculated using fc^{n+1} , according to Equation 2.

$$F_i = - \int_{solid} fc_i^{n+1} dV + \int_{solid} \left(\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} \right) dV \quad (2)$$

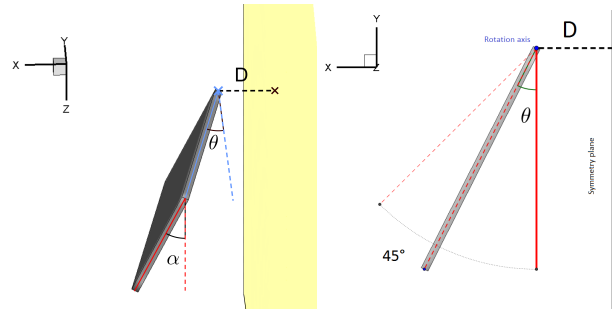


Figure 1: Three-dimensional view of the flapping motion.

This method displays spurious pressure fluctuations due to its method of dealing with the immersed boundary [23]. To reduce the effect of these fluctuations on the results, the force response of several flapping cycles will be averaged.

3 SIMULATION SETUP

To investigate how deformation of the chord affects the aerodynamic mechanisms around a flapping wing under hovering conditions, a simplified wing model has been created for the flow simulations. A wing will be modeled as a rectangular flat plate with a chord of $c = 0.1$ [m], a thickness of 0.006 [m] and an aspect ratio $AR = 2$. This aspect ratio value is at the lower side of the range of that of flying insects which is between 2 and 10 [24]. Instead of modeling two separate plates, a symmetry plane is used to reduce the required computational resources. Perspective and top views of the wing and the parameters defining the motion are shown in Figure 1. In this figure D represents the distance to the symmetry plane, and θ the flapping angle. Insects have a wide range of flapping angles [24] and in this case a total flapping angle θ of 45° is chosen. This flapping angle would be small for two-winged insects but is more typical to that of a four-winged MAV in biplane configuration [17]. The configuration would then correspond to one wing pair, with the assumption that the opposite wing pairs do not influence one another. The motion will start with an outstroke, which results in θ given by Equation 3.

$$\theta = 22.5 - 22.5 \cos(ft) \quad (3)$$

Where f is the flapping frequency and t is the elapsed time. The Reynolds number based on the average tip velocity and standard conditions at sea level is equal to $Re = 2002$, and a reduced frequency of $k = fc/U = 0.319$. All variables are summarized in Table 1. This table also includes the non-dimensionalized variables, which are calculated using the chord length and the average tip velocity. The different deformation scenarios considered in the simulations are detailed in the following subsections.

Parameter	Value	Dimensionless value
Chord	0.1[m]	1
Thickness	0.006[m]	0.06
Aspect ratio	2	2
Average tip velocity	0.314 [m/s]	1
Frequency	1 [s ⁻¹]	0.319
Reynolds number	2002	2002
Max flapping angle	45°	

Table 1: Parameters of the simulations

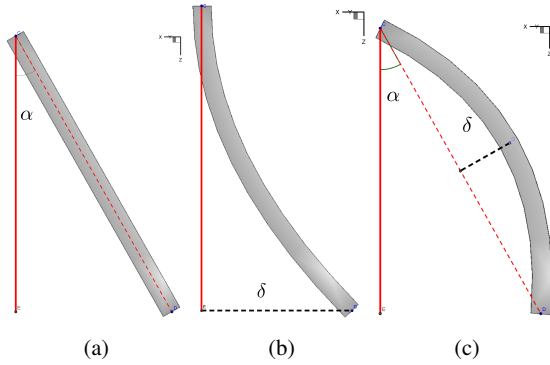


Figure 2: Chordwise plate deformation profiles. a) Rigid case; b) Deforming end case; c) Deforming center case

3.1 Rigid case with rotation

For the rigid plate case the flapping motion described in the previous section is combined with a pitching motion to create an angle of attack. This is done by rotating the plate around the leading edge as shown in Figure 2a. This rotation angle is given by Equation 4.

$$\alpha = -\alpha_{max} \sin(kT) \quad (4)$$

In which α_{max} represents the maximum angle. Note that this angle α is not the angle of attack as conventionally defined, but instead 90 degree minus the angle of attack.

3.2 Deforming end

The first deforming case does not contain any pitching, instead an angle of attack is created by the deformation alone. This is achieved by modeling that the leading edge remains straight and the maximum deformation occurs at the trailing edge. The shape of the chord is described using a quadratic function of the distance from the leading edge. The deformation is given by Equation 5.

$$\delta_{end} = -D_{end}^{max} \left(\frac{z}{c} \right)^2 \sin(kT) \quad (5)$$

Where D_{end}^{max} is the maximum deformation and $\frac{z}{c}$ is the distance from the leading edge normalized by the length of the chord. The resulting shape is shown in Figure 2b. It can

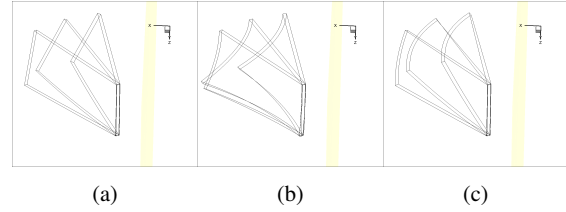


Figure 3: Three dimensional overview of the instrokes of the three cases. a) Rigid case; b) Deforming end case; c) Deforming center case

be seen that this shape increases the angle of attack near the leading edge compared to the rigid case.

3.3 Deforming center

The second deforming case allows the leading edge to rotate. It again uses a quadratic function to describe the deformation, with this case having the maximum deformation at the center of the chord. The deformation is given by Equation 6.

$$\delta_{center} = -D_{center}^{max} \left[1 - 4 \left(\frac{z}{c} - 0.5 \right)^2 \right] \sin(kT) \quad (6)$$

The deformation given by Equation 6 is combined with the angle of attack given by Equation 4 to give the complete deformed shape in time. The resulting cross-sectional shape of the wing is shown in Figure 2c. From the figure it is clear that for this case the deformation decreases the angle of attack at the leading edge compared to the rigid case.

In insect wings the deformation is usually higher near the wing tip due to torsional flexibility of the wing [25]. This is included in the present simulation model by varying the maximum deformation linearly along the span, resulting in a undeformed unrotated chord at the root of the plate and the maximum rotation and deformation at the tip. All deformation parameters are shown in Table 2. Three dimensional views of the instroke of the three cases are shown in Figure 3. Note in view of this spanwise deformation, that the rigid case indicates that there is no chordwise deformation, but it does not imply that the wing is rigid as a whole.

Parameter	Root value	Tip value
Angle α_{max}	0°	30°
D_{end}^{max}	0	0.5
D_{center}^{max}	0	0.175

Table 2: Deformation parameters.

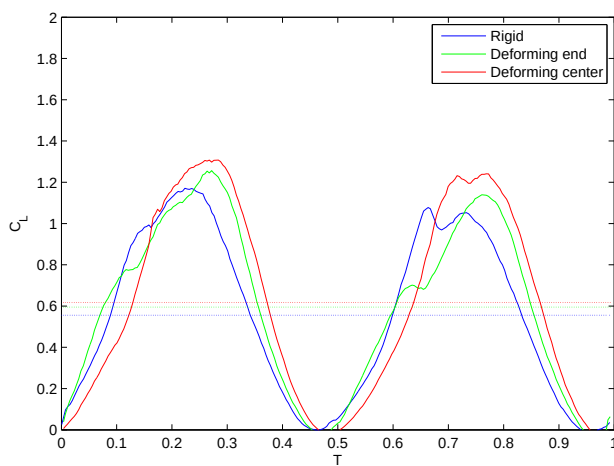
4 RESULTS

For the three cases the lift coefficients were calculated, and averaged over the last 3 cycles. Figure 4 shows the lift coefficient C_L , as well as the cycle average value. Here C_L is

Case	Average C_L
Rigid	0.5558
Deforming center	0.6171
Deforming end	0.5948

Table 3: Cycle averaged values of C_L for the cases.

defined as $C_L = \frac{L}{\frac{1}{2}\rho U^2 S}$ with lift L positive in negative z -direction. Although averaging the plots over 3 cycles reduces the fluctuations which are caused by the moving boundary, they are still present to some extent. Figure 4 represent one entire cycle, starting with an outstroke. The value of the cycle average C_L for the three cases is given in Table 3. It is observed that both deforming cases produce a higher average and peak lift than the rigid case.


Figure 4: C_L for the three cases. The motion starts with an outstroke.

A noticeable difference between the three cases is the C_L buildup at the beginning of both the instroke and the outstroke. The deforming end case shows the fastest C_L buildup, followed by the rigid case, and the deforming center case shows the slowest buildup. The difference in C_L buildup speed is caused by different LEV buildup speeds, and is related to the difference in leading edge orientation, as can be seen from Figure 5. This figure shows the Q criterion contour plots at half of the span, at the beginning of the outstroke. The deforming end case shown in Figures 5e and 5f clearly shows that the higher angle of attack in the deforming end case creates a stronger LEV compared to the rigid case (Figures 5a and 5b). The deforming center case shown in Figures 5c and 5d has a lower angle of attack and can be seen to create a weaker LEV.

The second clear difference between the cases is the distinct dip in C_L that can be seen in Figure 4 during the acceleration phase of the plates. This dip occurs both during the out-

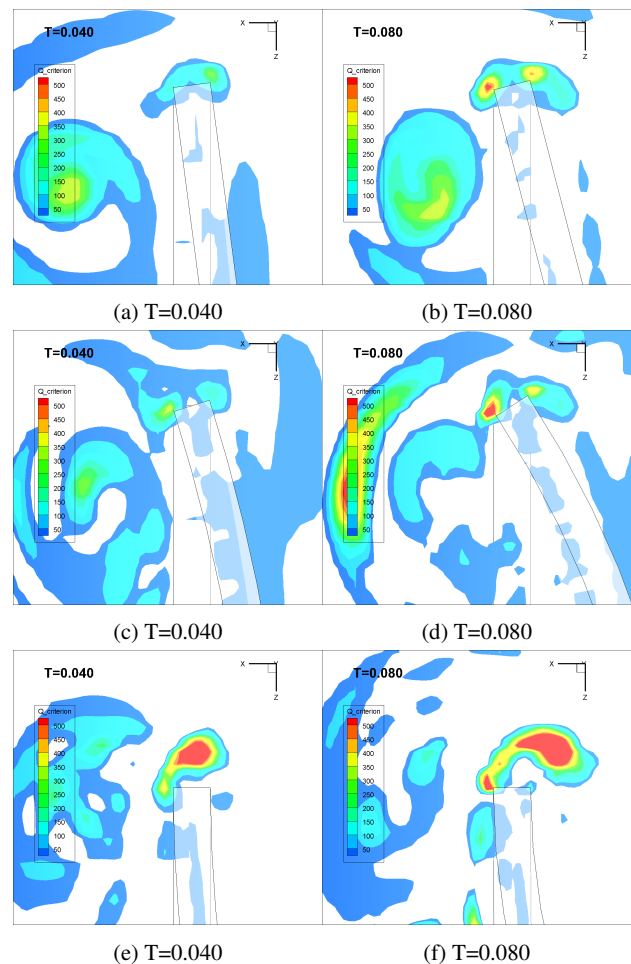


Figure 5: Q criterion contour plots of the three cases at the start of the outstroke with the plate moving to the left. Top: rigid case; middle: deforming center; bottom: deforming end

stroke and the instroke. The contour plots of the Q criterion are investigated at the moments around this dip during the instroke, at $T=0.559$, 0.599 and 0.639 at 50% of the span to see if this is associated to shedding of the LEV, to which such a decrease in lift buildup is commonly associated. The plots are shown for the deforming center and deforming end cases in Figure 6. Although it can be seen that the LEV moves further away from the plate for the deforming end case, the difference does not explain the large dip shown in C_L . Contour plots of the Q criterion at different locations along the span do not show distinct shedding of the LEV either. Since the shedding of the LEV cannot be identified as the source for the dip, the pressure isosurfaces are investigated as further means to visualize the vortex behaviour around the wings. The pressure isosurfaces with $p = -1$ at the beginning of the instroke are shown in Figure 7. In Figures 7a and 7b the vortices from the previous outstroke stroke can be clearly identified, with the shed tip vortex, trailing edge vortex and LEV in front of the plate. Figure 7c shows that at $T=0.64$, the new LEV in the deforming end case interacts with the previously shed LEV, this instant corresponds to the dip in C_L in Figure 4. The deforming center case shows no interaction, and no change in the C_L slope as seen in Figure 4. At the next instance, shown in Figure 7e the shed LEV from the previous stroke has been completely absorbed into the new LEV for the deforming end case, and C_L is again increasing. For the deforming center case the dip occurs much later, as does the interaction between the previously shed vortex and the new LEV.

From the previous results it can be concluded that not the shedding of the LEV is responsible for the dip in C_L seen in Figure 4, but the interaction of the LEV with the shed LEV from the previous stroke. For this to be true there should be no dips in the buildup of C_L during the initial stroke, since there will be no previously shed vortices in this case, and investigation shows that these dips are indeed absent [23]. The faster buildup of the LEV in the deforming end case causes the interaction with the shed vortex to occur earlier. The lowering of the angle of attack by the deforming center case delays the interaction and reduces the negative effect. The curvature of the chord of the plate in the deforming center case also causes the previously shed vortex to get trapped below the plate, delaying the interaction further. This can be clearly seen in Figure 8.

To investigate the effect of the deformation on the clap and fling mechanism the distance to the symmetry plane was varied. Three cases are defined with $D=0.25c$, $0.50c$ and a case without symmetry plane. Table 4 shows the increase in the average C_L compared to the case without symmetry plane, for both the rigid case and the deforming center case. Although the average C_L increases for both cases when the symmetry plane is closer an important difference can be noted. The increase in average C_L is higher for the deforming center case with the symmetry plane at $D=0.50c$, but higher for the rigid case with the symmetry plane at $D=0.25c$. The

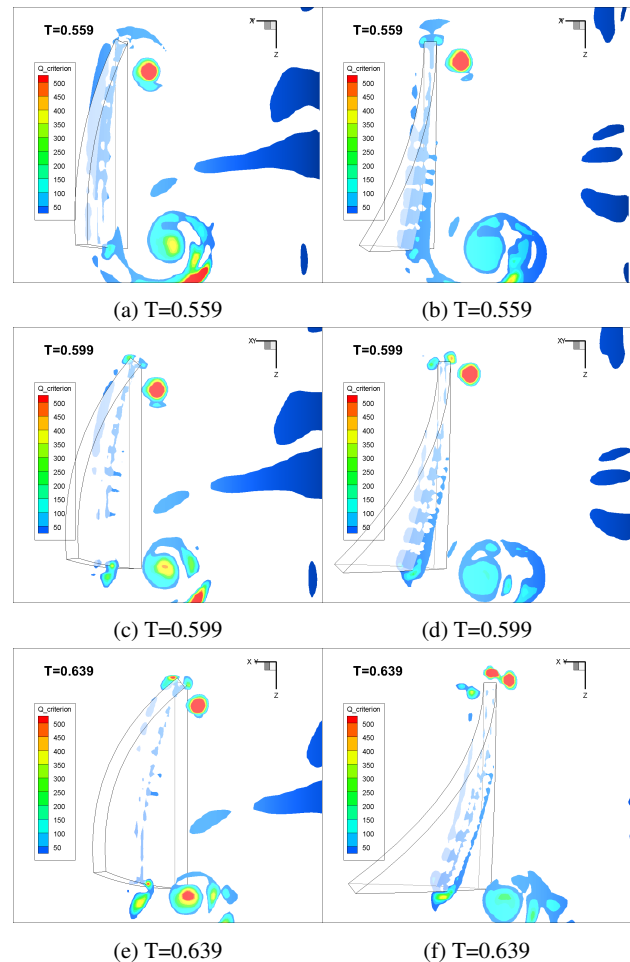


Figure 6: Q criterion contour plots of the deforming end and deforming center cases during the instroke, with the wings moving to the right. The deforming center case on the left and the deforming end case on the right.

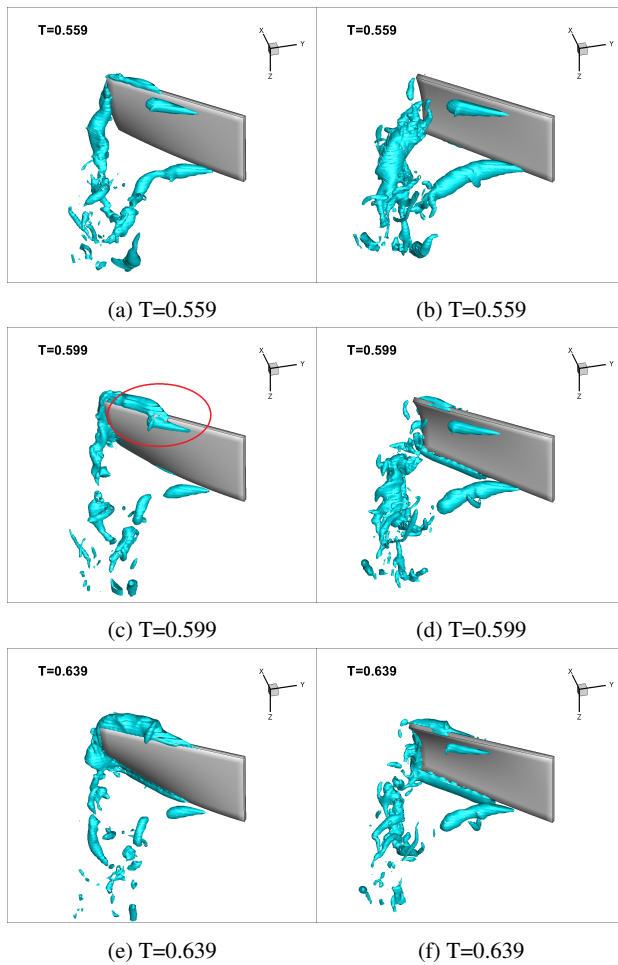


Figure 7: Pressure isosurfaces of $p = -1$ at the starting of the instroke with the plate moving to the front. Left for the deforming end case and right for the deforming center case.

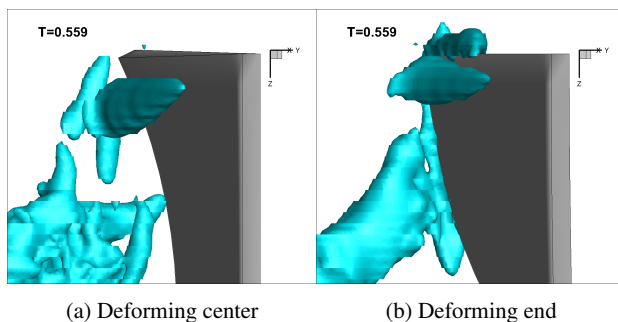


Figure 8: Previously shed vortices at the beginning of the instroke for the the deforming center and deforming end cases.

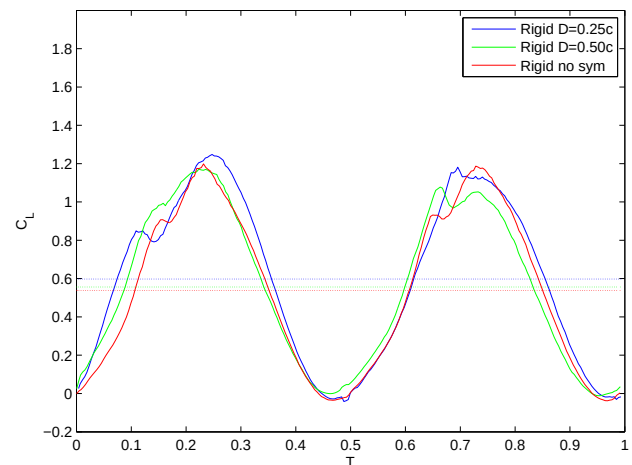


Figure 9: Lift coefficient plot for the rigid cases, at different distances from the symmetry plane. The motion starts with an outstroke.

difference between the cases can be explained by looking at the development in C_L as shown in Figures 9 and 10. Figure 10 clearly shows a very large dip in C_L buildup during the outstroke for the case with $D=0.25c$. This dip is much larger than for the case with the symmetry plane at $D=0.50c$, and greatly reduces the average C_L . The faster LEV buildup caused by the presence of the symmetry plane causes the interaction between the LEV and the previously shed vortex to occur sooner. Since the previously shed vortex is trapped below the wing as was shown in Figure 8b, the interaction takes significantly longer than for the rigid case. This longer interaction limits the benefit of fast LEV buildup.

	0.5c	0.25c
Rigid case	+3.4%	+11.3%
Deforming center case	+5.1%	+9.2%

Table 4: Increase in cycle average C_L by changing distance to symmetry plane.

5 CONCLUSION

Three different deforming wings were simulated to investigate the effect of the deformation on lift generation by unsteady aerodynamic mechanisms. It was shown that deforming the wing in such a way that the angle of attack near the leading edge increased, leads to faster LEV buildup and with this a faster buildup of lift. No shedding of the LEV was observed in any of the cases even when deformation caused a high angle of attack near the leading edge. The distinct dip in lift buildup that was observed was shown to be caused by the interaction between the previously shed vortex and the newly developing LEV. This interaction occurred faster when the angle of attack near the leading edge was increased, and

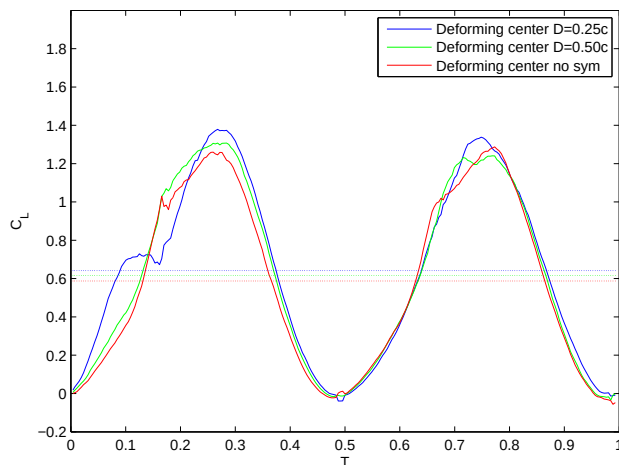


Figure 10: Lift coefficient plot for the deforming center cases, at different distances from the symmetry plane. The motion starts with an outstroke.

slower when the angle of attack was decreased. The lower angle of attack caused the previously shed vortex to get trapped below the wing preventing the interaction. Moving the wing closer to the symmetry plane had a positive effect on the cycle average value of C_L . This positive effect was reduced however by the earlier interaction between the LEV and the previously shed vortex. This was especially clear for the case where the angle of attack was reduced, since the position of the previously shed vortex caused the interaction to take longer.

REFERENCES

- [1] S. A. Ansari, N. Phillips, G. Stabler, P. C. Wilkins, R. Zbikowski, and K. Knowles. Experimental investigation of some aspects of insect-like flapping flight aerodynamics for application to micro air vehicles. *Experiments in Fluids*, 46:777–798, 2009.
- [2] R. Zbikowski. On aerodynamic modeling of an insect-like flapping wing in hover for micro air vehicles. *Philosophical Transactions of the Royal Society of London A*, 360:273–290, 2002.
- [3] R. B. Srygley and A. L. R. Thomas. Unconventional lift-generating mechanisms in free-flying butterflies. *Nature*, 420:660–664, 2002.
- [4] C. P. Ellington, C. van den Berg, A. P. Willmott, and A. L. R. Thomas. Leading-edge vortices in insect flight. *Nature*, 384:626–630, 1996.
- [5] M. H. Dickinson, F. O. Lehmann, and S. P. Sane. Wing rotation and the aerodynamic basis of insect flight. *Science*, 284, 1999.
- [6] F.-O. Lehmann and S. Pick. The aerodynamic benefit of wing-wing interaction depends on stroke trajectory in flapping insect wings. *The Journal of Experimental Biology*, 210:1362–1377, 2007.
- [7] L. David, T. Jardin, P. Braud, and A. Farcy. Time-resolved scanning tomography piv measurements around a flapping wing. *Experiments in Fluids*, 52:857–864, 2012.
- [8] D. Kim, F. Hussain, and M. Gharib. Vortex dynamics of clapping plates. *The Journal of Fluid Mechanics*, 714:5–23, 2013.
- [9] Z. Wang. Two dimensional mechanism for insect hovering. *Physical Review letters*, 85(10):2216–2219, 2000.
- [10] B. Yin and H. Luo. Effect of wing inertia on hovering performance of flexible flapping wings. *Physics of Fluids*, 22(11):1071–1080, 2010.
- [11] J. D. Eldredge, J. T., and A. Medina. On the roles of chordwise flexibility in a flapping wing with hovering kinematics. *Journal of Fluid Mechanics*, 659:94–115, 2010.
- [12] T. Gillebaart. Influence of flexibility on the clap and peel movement of the DelFly II. Master's thesis, Delft University of Technology, 2011.
- [13] J. R. Usherwood and C. P. Ellington. The aerodynamics of revolving wings I. model hawkmoth wings. *The Journal of Experimental Biology*, 205:1547–1564, 2002.
- [14] J. A. Walker. Rotational lift: something different or more of the same? *The Journal of Experimental Biology*, 205:3783–3792, 2002.
- [15] T. Jardin, A. Farcy, and L. David. Three-dimensional effects in hovering flapping flight. *The Journal of Fluid Mechanics*, 702:102–125, 2012.
- [16] P. Trizila, C.K. Kang, H. Aono, and W. Shyy. Low-reynolds-number aerodynamics of a flapping rigid flat plate. *AIAA Journal*, 49(4):806–824, 2011.
- [17] K. M. E. de Clercq, R. de Kat, B. Remes, B. W. van Oudheusden, and H. Bijl. Aerodynamic experiments on delfly II: Unsteady lift enhancement. *International Journal of Micro Air Vehicles*, 1(4):217–224, 2009.
- [18] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161:35–60, 2000.
- [19] J. Kim, D. Kim, and H. Choi. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics*, 171:132–150, 2001.

- [20] J. Yang and E. Balaras. An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *Journal of Computational Science*, 215:12–40, 2006.
- [21] C. C. Liao, Y. W. Chang, C. A. Lin, and J.M. McDonough. Simulating flows with moving rigid boundary using immersed-boundary method. *Computers & Fluids*, 39:152–167, 2010.
- [22] W.B. Tay, B. W. van Oudheusden, and H. Bijl. Numerical simulation of x-wing type biplane flapping wings in 3d using the immersed boundary method. *Bioinspiration and Biomimetics*, 9:1–21, 2014.
- [23] T. A. Noyon. The effect of wing deformation on the unsteady aerodynamic mechanisms in hovering flapping flight. Master's thesis, Delft University of Technology, 2014.
- [24] R. Dudley. *The Biomechanics of Insect Flight: Form, Function, Evolution*. Princeton University Press, reprint edition, 2002.
- [25] K. Mazaheri and A. Ebrahimi. Experimental investigation of the effect of chordwise flexibility on the aerodynamics of flapping wings in hovering flight. *Journal of Fluids and Structures*, 26:544–558, 2010.

Pitch and roll control mechanism for a hovering flapping wing MAV

M. Karásek*, A. Hua, Y. Nan, M. Lalami, and A. Preumont
Université libre de Bruxelles, Belgium

ABSTRACT

Hovering flapping flight is inherently unstable and needs to be stabilized actively. We present a control mechanism that modulates independently the wing flapping amplitude and offset by displacing joints of a flapping linkage mechanism. We demonstrate its performance by high speed camera recordings of the wing motion as well as by direct measurements of pitch moment and lift force. While flapping at 17 Hz the prototype produces 90 mN of lift and generates pitch moments from -0.7 N.mm to 1.1 N.mm. The mechanism shows low level of cross-coupling in combined pitch and roll commands.

1 INTRODUCTION

Flapping wing Micro Air Vehicles (MAVs) take the inspiration from nature. They mimic insects and hummingbirds who, unlike other larger birds, flap their wings in approximately horizontal plane, which allows them to hover for extended periods of time. Since the flapping flight of two winged animals is inherently unstable [1, 2, 3, 4], the animals stabilize their attitude by small changes of the wing angle of attack, of mean wing position and of flapping amplitude [5].

The stability can be augmented by an artificial increase of aerodynamic damping [6]. Passively stable MAVs use tails and sail-like damping surfaces [7, 8, 9], but stay sensitive to external disturbances. Tail-less flapping wing MAVs are unstable and, like their natural counterparts, need to be stabilized actively. Nevertheless, this also makes them very agile when manoeuvring. The active stabilization requires a feedback system that senses the changes in the MAV attitude and actively produces stabilizing moments around the pitch, roll and yaw axes.

Two strategies of moment generation were used in the existing MAVs. The Nano Hummingbird [10] generates the moments by three servos that modulate the wing twist, similar to the changes of angle of attack in insects. The second strategy, used by the Harvard robotic fly [11], generates the moments by modulating the wing flapping amplitude, mean position and difference of speed velocities in upstroke and downstroke. Two piezoactuators (PZTs) are used for propulsion and the control is implemented by shaping the driving signals for the PZTs.

*matej.karasek@ulb.ac.be, <http://scmero.ulb.ac.be>

In this paper we present a control mechanism based on flapping amplitude and offset (mean wing position) modulation, see Figure 1. A novel solution is used to modify the kinematics of the flapping linkage mechanism by displacing its joints by servo motors. We describe the concept kinematics and prototype design and present experimental results obtained by high speed camera and force balance measurements.

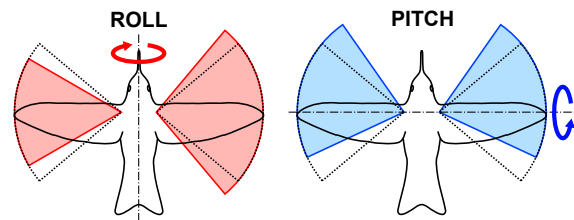


Figure 1: Control via flapping amplitude & offset modulation.

2 FLAPPING MECHANISM

The flapping mechanism was already presented in [12], see Figure 2. A linkage mechanism, driven by a DC motor, is used to transform the motor shaft rotation into the flapping motion of the wings. It has two stages: a slider crank based mechanism generates a low amplitude oscillating motion and a four-bar linkage amplifies the motion to the desired amplitude $\Phi = 120^\circ$ (Figure 3). The slider crank mechanism is common for the two wings. The advantage of using two stages is that the asymmetries coming from individual stages can be compensated by an appropriate choice of dimensions.

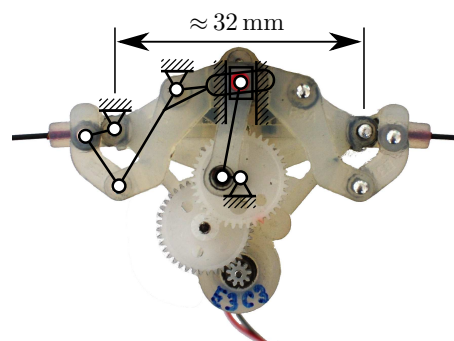


Figure 2: Flapping mechanism prototype and its kinematic representation.

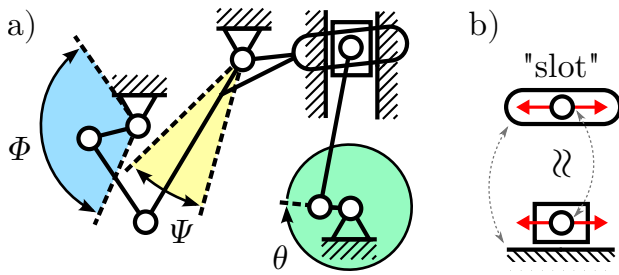


Figure 3: a) Two stages of the flapping mechanism: slider crank with a rocker producing oscillating motion and a four-bar mechanism for motion amplification. b) The symbol for a "slot" kinematic pair (equivalent to a slider).

2.1 Kinematics

The kinematics of the proposed mechanism can be treated separately for each mechanism stage. Using the notation of Figure 4 the kinematics of the first stage can be expressed analytically as

$$\psi = \arctan \left(\frac{A_1 - L_1 \cos \theta - \sqrt{L_2^2 - L_1^2 \sin^2 \theta}}{L_3} \right) + \frac{\pi}{2} \quad (1)$$

where θ is the input angle and ψ is the angle of the intermediary link 3-4. The analytic solution of the second (amplification) stage is a classical solution of a four bar mechanism

$$\phi = \arctan \left(\frac{a}{b} \right) - \arccos \left(\frac{c}{\sqrt{a^2 + b^2}} \right) \quad (2)$$

where

$$\begin{aligned} a &= -2L_4L_6 \sin(\psi - \alpha) \\ b &= 2A_2L_6 - 2L_4L_6 \cos(\psi - \alpha) \\ c &= L_5^2 - A_2 - L_4^2 - L_6^2 + 2A_2L_4 \cos(\psi - \alpha) \end{aligned} \quad (3)$$

The dimensions were optimized numerically for the desired amplitude as well as for symmetry of upstroke and downstroke velocity profiles. The final dimensions in Table 1 result into nearly harmonic flapping motion. The final structure of the flapping mechanism was rearranged (compared to Figure 4) to minimize the overall mechanism dimensions. The mutual orientation of the two stages was adapted as in Figure 2 and the wing bar was connected to the output link at an angle in order to obtain symmetric motion with respect to the lateral body axis.

2.2 Mechanism design

The frame and the links of the flapping mechanism are all 3D printed by PolyJet technology in Digital ABS photopolymer. Thanks to very fine resolution ($42 \mu\text{m}$ in xy, $16 \mu\text{m}$ in z - Objet Eden series) the parts do not require any further

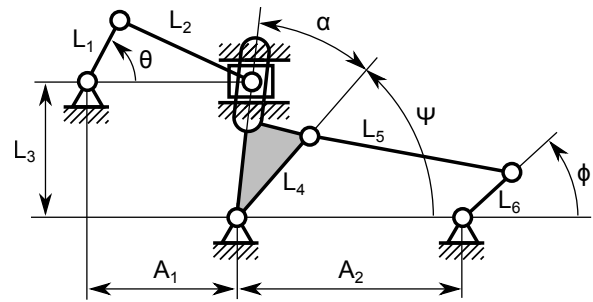


Figure 4: Kinematic scheme of the flapping mechanism.

Table 1: Mechanism dimensions (lengths in mm, angles in °)

L_1	L_2	L_3	L_4	L_5	L_6	A_1	A_2	α
2.25	12	8	14	7.53	3.57	11	-9.2	-70

processing. Aluminium and steel rivets are used to connect the links together. The shoulder hinges have integrated brass bearings to increase their robustness.

The motor is placed above the flapping mechanism, in the centre between the two wing shoulders. A two stage gear-box (reduction 1:19.75) is used to reduce the motor speed. The most recent version of the flapping mechanism weights 6.9 g without the motor. With a 5.2 g brush-less DC motor (Faulhaber 0824) and 90 mm wings it produces $155 \text{ mN} \approx 15.8 \text{ g}$ lift force while flapping at 24 Hz. The electrical power consumption is 4W.

2.3 Wings

The design of the wings is inspired by the Nano Hummingbird [10]. The wing is made of a 15 micron thin polyester membrane. It has two sleeves, one on the leading edge and one on the root edge (close to the body), that accommodate the leading edge and root edge carbon-fibre-reinforced polymer (CFRP) bars. The sleeves can rotate freely around the bars and are reinforced with Icarex for durability. Since the angle between the sleeves is greater than the angle between the bars the wing becomes cambered and twisted after the assembly (Figure 5). The camber is bistable - it flips passively from one side to another depending on the direction of motion. The root bar is fixed directly to the frame

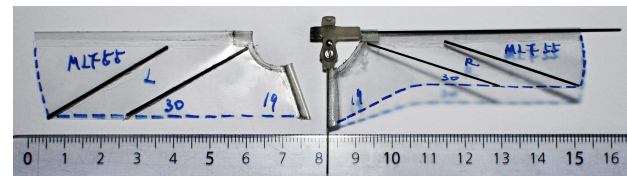


Figure 5: Polyester film wing becomes cambered after assembly.

and is aligned with the shoulder axis, the leading edge bar is connected to the output link of the flapping mechanism.

The wings are hand built - a reasonable accuracy and repeatability is achieved by printing the desired shape on a sheet of paper that is attached under the membrane and used as a template for cutting. 1 mm x 0.12 mm CFRP bands are used as stiffeners. The placement of stiffeners, the overall shape, the wing length, the surface, the aspect ratio or the angle between the sleeves were optimized through experiments.

3 CONTROL MECHANISM

The control mechanism is the second, but equally important part of the hummingbird prototype. Its role is to generate moments for attitude stabilization and flight control. In our previous work [12] we presented a concept inspired by the Nano hummingbird [10] that produces the moments by modulating the wing twist. This concept, however, requires specific wing design. The produced lift must increase when the membrane is stretched (the twist decreased) and decrease when the membrane is loosened (the twist increased). Thus, at nominal conditions the wing needs to be operated

below its maximal lift.

The alternative concept presented here generates the control moments by modulating the flapping amplitude and offset (mean wing position). Compared to wing twist modulation it is more intuitive and works with any wing design.

3.1 Amplitude and offset modulation via joint displacement

Stabilizing moments can be generated by amplitude and offset modulation as in Figure 1. The roll moment is produced simply by increasing the amplitude of one wing and decreasing the amplitude of the other wing. Moving the mean wing position of both wings forwards or backwards results in a nose-up or nose-down pitch moment, respectively. Finally the yaw moment can be generated by changing the mean wing position asymmetrically.

We have found that both amplitude and offset can be controlled by displacing the mechanism joints marked in Figure 6 a) with orange arrows. The maps in Figure 6 b) show the relation between the position of the joints and the wing amplitude Φ and offset ϕ_0 . The blue lines connect positions with constant amplitude and the red lines connect positions

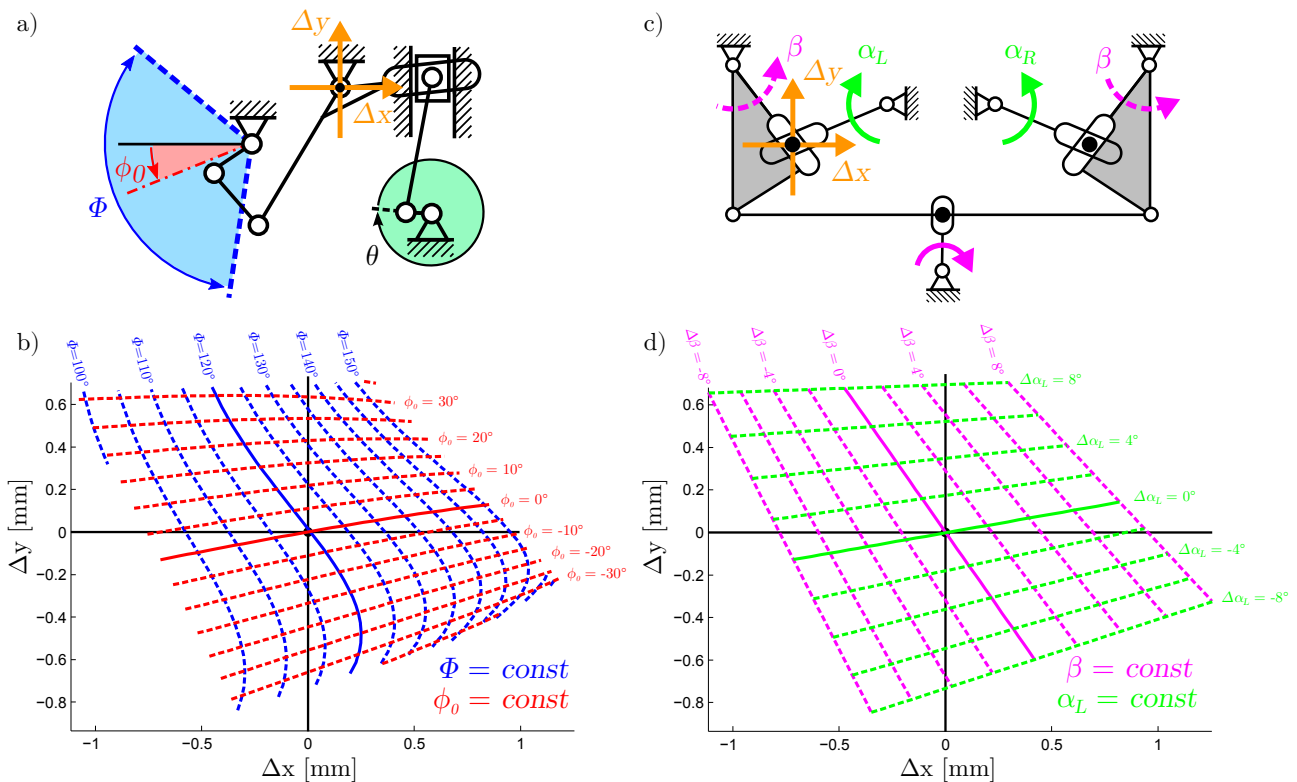


Figure 6: Amplitude and offset modulation via joint displacements: a) the flapping mechanism and its joint displaced by $[\Delta x, \Delta y]$, b) lines of constant amplitude Φ and offset ϕ_0 for varying joint position, c) the control mechanism with 3DOF ($\alpha_L, \alpha_R, \beta$) defining the joint position by an intersection of two channels, d) the lines of constant commands α_L and β that approximate well the lines of constant amplitude and offset. Only the plots for the left wing are displayed, the line plots for the right wing are the same but mirrored.

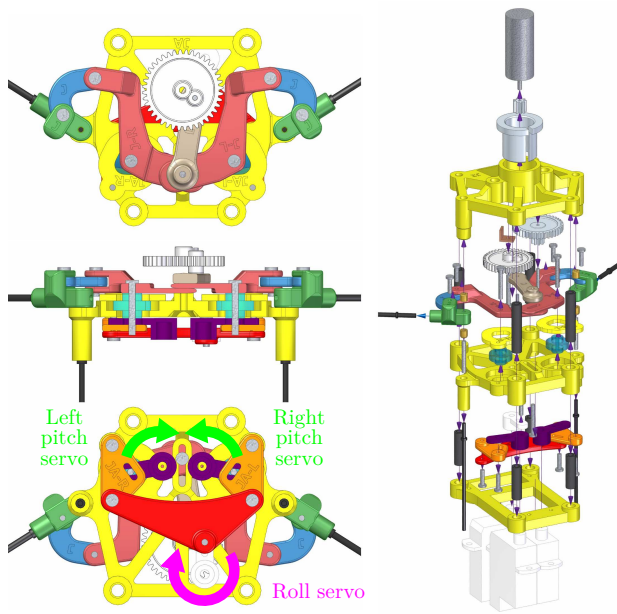


Figure 7: Control mechanism: bottom (left), section (right) and top view on the flapping mechanism with the joint displacement system.

with constant offset. Thus, moving the joint along a blue line will modify the offset, but the amplitude will remain constant. Similarly a displacement along a red line will only affect the amplitude.

It can be noticed that the two sets of curves cross each other at high angles (above 70°) which means the two values can be controlled independently. Moreover, the lines of constant offset are almost straight and nearly parallel; the curves of constant amplitude are also equally spaced and can be approximated by straight lines around the nominal position. This allows to design a joint displacement mechanism with two degrees of freedom (DOF) where the control is decoupled - one DOF controls directly the amplitude and the other controls the offset.

3.2 Control mechanism prototype

The scheme of the proposed mechanism for joint displacement is in Figure 6 c). Each joint is displaced by two links with slots that rotate with respect to the frame by angles α and β , respectively. The joint position is defined by an intersection of the two slots. The link hinges are located on the lines of nominal amplitude and of the nominal offset, respectively. If one of the links is blocked and the other one is moving, the joint moves along a line defined by the slot of the blocked link. If the hinges are placed far enough from the nominal position of the displaced joint ($\Delta x = 0$, $\Delta y = 0$), these lines appear nearly parallel in the region of interest and the joint paths approximate well the theoretical curves of con-

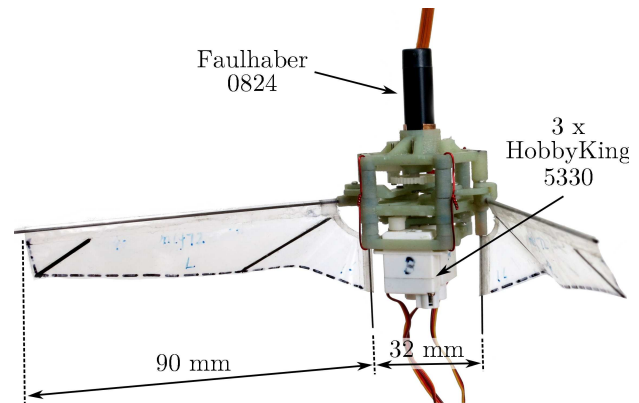


Figure 8: The assembled prototype.

stant amplitude and offset Figure 6 d).

The control of left and right wing offset needs to be independent, operated by separate actuators: a symmetric offset change ($\Delta\alpha_L = \Delta\alpha_R$) will produce pitch moment while asymmetric change ($\Delta\alpha_L = -\Delta\alpha_R$) will produce yaw moment. However, the amplitude can be controlled by a single actuator (β): only asymmetric amplitude changes are needed for roll. This can be achieved by a parallelogram linking the two links responsible for amplitude control (Figure 6 c)).

The final mechanical solution of the joint displacement mechanism is in Figure 7. The rivets of the joints to be displaced are fixed from the top to "anchors" that are free to slide in the horizontal plane of the frame. The displacement is limited to the zone considered in Figure 6 d) by the shape of the frame cut-out. All the parts were 3D printed, a photograph of an assembled prototype is in Figure 8. The mechanism is actuated by three micro servos (HobbyKing 5330) with a weight of 2 g each. The total weight of the controlled prototype including the servos (6 g) and the propulsion motor (5.2 g) is 21.4 g.

4 EXPERIMENTS

The experiments presented here include high speed camera measurements of the wing kinematics as well as lift and pitch moment measurements on a custom build force balance [12]. Unless mentioned otherwise, all the tests were carried out at a moderate flapping frequency of 15Hz in order to have consistent results in all the tests. For higher frequencies the prototype performance can slightly deteriorate over time due to wear.

4.1 Wing kinematics

To test the function of the control mechanism we used a Photron FASTCAM SA3 high speed camera (resolution 1024 x 1024 pixels). We recorded the prototype wing motion at 500 fps under different control commands and we tracked the sweep angle ϕ at the wing tips according to Figure 9. The amplitude was subsequently calculated as $\Phi = \phi_{max} - \phi_{min}$.

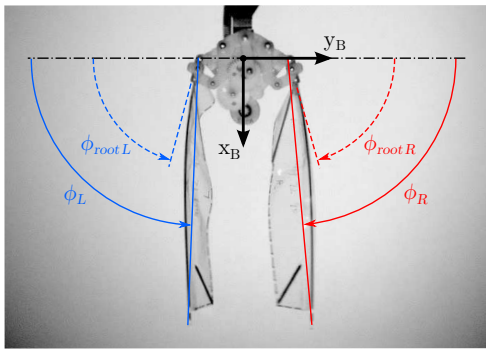


Figure 9: Definition of the tracked angles.

and offset as $\phi_0 = (\phi_{max} + \phi_{min})/2$, where ϕ_{max} and ϕ_{min} are the maximal and minimal observed angles ϕ , respectively. The observed amplitudes were much larger than the design value of 120° due to compliance of the wing bars and partly also due to mechanism backlash.

Figure 10 shows the results of amplitude difference control. The servos controlling the wing offset were kept in their nominal position and the roll control servo was commanded from the minimal to the maximal position with a step of 10% of the full range. The left and right wing amplitudes, Φ_L and Φ_R , are approximately equal for zero servo position. As intended, their difference $\Delta\Phi = \Phi_R - \Phi_L$ increases/decreases approximately linearly as the servo moves towards the positive/negative limit, where the difference is $+52^\circ$ and -44° , respectively. Thus, a good control authority of the roll moment is achieved. The wing offset remains relatively close to zero and the average amplitude $\bar{\Phi} = (\Phi_R + \Phi_L)/2$ stays approximately constant (around 155°) in the central part of the servo range. There exist some imperfections, in particular close to the limits, but these should be compensated by feedback control in future.

The offset control is presented in Figure 11. The left and right offset servos were commanded together over the full

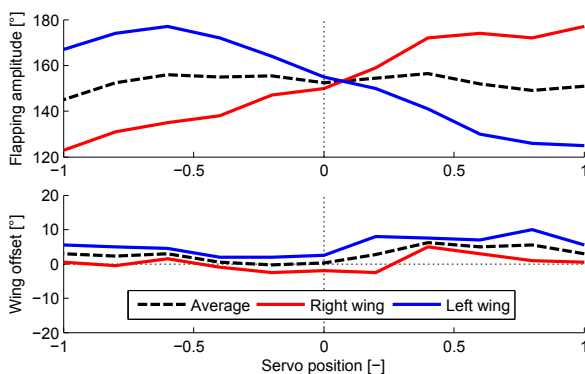


Figure 10: Amplitude control with the roll servo.

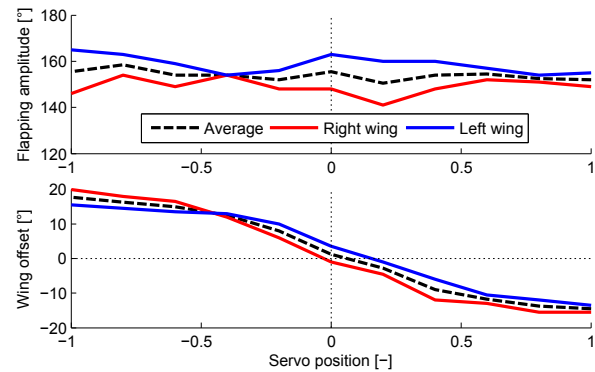


Figure 11: Offset control with the pair of pitch servos.

range, again with a step of 10%. The roll servo was kept at zero. We again see a good control authority over the wing offset, being linear around the origin with a slight decrease of slope closer to the limits. The maximal and minimal average offset $\bar{\phi}_0 = (\phi_{0R} + \phi_{0L})/2$ is $+17.7^\circ$ and -14.5° , respectively. The amplitude of the left and right wing varies quite a lot, but the average $\bar{\Phi}$ stays close to 155° . A simultaneous control of the amplitude difference would be necessary to achieve zero roll moment. We will discuss the combined commands and resulting coupling effects at the end of this section.

4.2 Control mechanism dynamics

Figures 12 and 13 show the dynamics of the transition from minimal to maximal command of pitch and roll, respectively. For this experiment the flapping frequency was 17Hz. The figures display the wing tip angles, their extremal positions are connected with a dashed line and the average position (over the last wingbeat) is displayed as dash-dotted line. An LED was placed on the prototype to indicate the moment of the step command (black line).

As can be seen in Figure 12 the transition from maximal to minimal offset occurs within 2 wingbeats. The transition from negative to positive amplitude difference takes around 4 wingbeats (Figure 13), however an opposite sign of the difference is achieved already for 2 wingbeats. The same could be observed for the step commands in opposite directions.

We have two explanations for the faster pitch command. First, in offset control two servos are employed, one acting on each joint, while in the roll command only a single servo is displacing the two joints. The second reason is that the reaction due to the flapping motion on the displaced joints has a major component in the direction, where the joints are displaced for offset control. This speeds up the offset transition when a change is desired, but has an adverse effect on the mechanism efficiency as the joints keep shaking in this direction during operation.

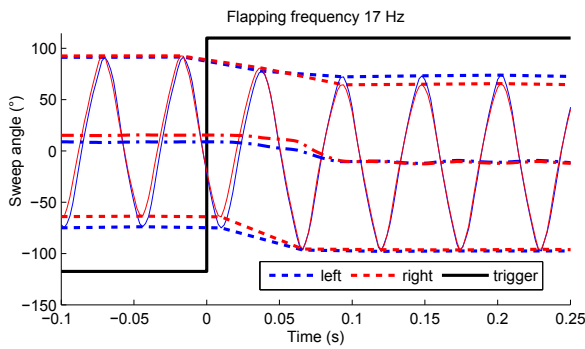


Figure 12: Pitch up → down command dynamics

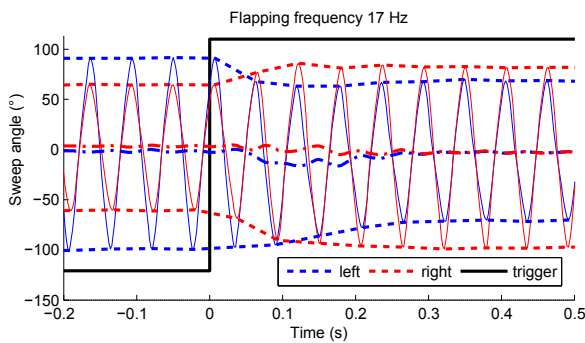


Figure 13: Roll left → right command dynamics

4.3 Pitch moment and lift generation

Apart from wing kinematics measurements the performance of the prototype was tested directly on a force balance. The custom built balance was presented in [12]. It was designed to measure cycle averaged lift and pitch moment with a help of two double cantilever beams with strain gages.

Figure 14 shows the measured pitch moment, lift, frequency and motor current as a function of the offset servos position. We kept the motor voltage at 4.2V, at which the flapping frequency is approximately 15Hz for the nominal servo position. The measured pitch moment ranged from -0.5 N.mm to 0.8 N.mm. The lift force and frequency increase and the current decreases when the servos approach the limit positions. This is caused by the shaking of the displaced joints due to flapping (already mentioned earlier) that happens particularly for the central servo positions. The joints get a better fix in the limit positions, where the servo pushes the displaced joints against a wall of the frame, and thus the efficiency increases. The lift varies between 53 and 59 mN and the frequency between 14.9 and 16.2 Hz, which fits the lift vs frequency characteristic of the uncontrolled prototype (joints are fixed). However, the electrical power of the controlled version is almost twice as high due to losses in the shaking joints.

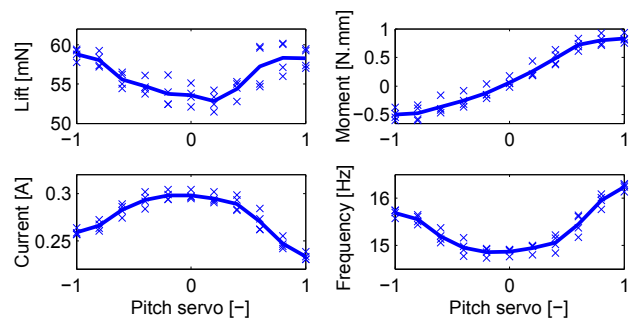


Figure 14: Lift, pitch moment, motor current and flapping frequency against pitch servos position. Measurement done at 15 Hz, individual measurements are displayed as crosses, the solid line represents an average of four measurements.

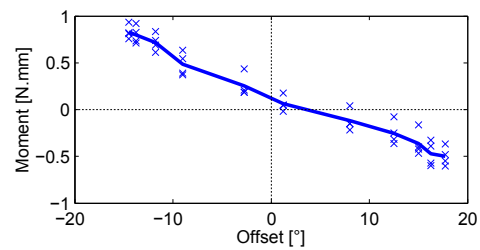


Figure 15: Pitch moment against wing offset.

In Figure 15 we combine the wing kinematics measurement from the previous part with the moment measurement. The relationship is approximately linear with a slope of 0.04 N.mm per degree. The non-zero moment produced at zero offset can be explained by a combination of asymmetric wing design (stiffeners glued only from one side), different velocity profiles in upstroke and downstroke and by an imperfect alignment of the prototype on the balance. Nevertheless, a compensating moment can be easily introduced by an offset of the COG from the shoulders in the longitudinal direction.

Figure 16 shows the pitch moment, lift, frequency and motor current for motor voltages up to 6V. We plot three curves for pitch servos at positions -1, 0 and 1, related to the full range. The behaviour corresponds to the one observed at 15Hz. At the highest tested voltage the mechanism produces pitch moments from -0.7 N.mm to 1.1 N.mm while the lift ranges between 90 and 100 mN.

4.4 Combined commands

Finally, we tested combinations of pitch and roll commands to identify the amount of cross-coupling. Again all the measurements were carried out at a constant motor voltage (4.2V) giving a flapping frequency of around 15Hz at the nominal servos position. The measurements were taken at servo positions -0.8, -0.4, 0, 0.4 and 0.8 of the full range for both pitch and roll servos. Thus, 25 measurements were taken in total.

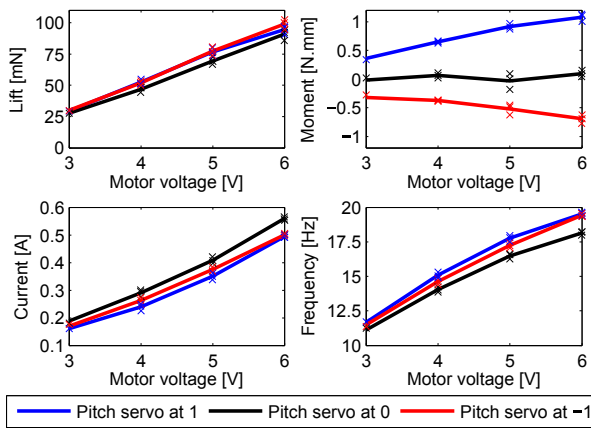


Figure 16: Lift, pitch moment, motor current and flapping frequency measured for increasing motor voltage. Black lines represent the zero pitch servos position, red and blue lines represent the minimal and maximal pitch command. The crosses represent individual measurements.

Figure 17 shows the amplitude and offset maps of the left and the right wing as obtained from the high speed recordings. We can see that while small cross-coupling always exists, moving the roll servo has a dominant effect on the amplitude and the pitch servo has a dominant effect on the offset. Moreover, the relation between the amplitude/offset and roll/pitch servo positions stays always monotonic. Thus, a feedback controller should be able to compensate the coupling effects and the small differences between left and right wing behaviour, caused by the mechanism imperfections.

The same experiment was repeated with the force balance, the results are in Figure 18. The maps show the pitch moment and lift force maps for the servo input combinations considered. The lift map keeps a valley-like shape in the pitch servo direction, similar to what was observed for the pure pitch command and what can be explained by improper joint fixation in the central pitch servo positions. There is also a smaller increase of lift in the positive roll direction, which we believe comes from the imperfections of the prototype. The minimum and maximum lift is 48 and 59 mN, respectively. The lift force variation is related to the square of the frequency times amplitude in Figure 17 c).

The pitch moment depends mostly on pitch servo positions, while there are only minor differences when the roll servo position changes. Thus, the minimal and maximal pitch values stay at the same levels as for the pure pitch command, -0.4 N.mm and 0.8 N.mm respectively. The pitch moment corresponds closely to the mean offset in Figure 17 f).

5 CONCLUSION

We presented a control mechanism for a flapping wing MAV that generates necessary control moments by modulating the flapping wing amplitude and offset. A novel con-

cept of joint displacement is used to modify the kinematics of the linkage mechanism producing the flapping motion. We demonstrated experimentally that sufficient offset ($\pm 15^\circ$) and amplitude differences (above $\pm 40^\circ$) for pitch and roll control can be introduced by very small displacements of the linkage joints (below $\pm 1\text{mm}$) in two directions. The transitions between maximum and minimum command takes two wingbeats in pitch and up-to four wingbeats in roll. Moreover, very low level of cross-coupling exists for combined commands. The prototype can produce pitch moments between -0.7 N.mm and 1.1 N.mm while flapping at frequencies around 18Hz and producing a lift of at least 90mN.

While the control mechanism succeeds in modifying the wing kinematics, the prototype efficiency drops significantly compared to the uncontrolled prototype. The drawback of the proposed solution is that the displaced joints need to hold rather large and oscillating reaction forces due to flapping. This causes the joints to shake, which reduces significantly the mechanism performance and at higher frequencies also its lifespan. The motor draws up to twice the electrical power compared to an uncontrolled prototype with fixed joints. We are working on an alternative mechanical solution that should reduce the effect of the oscillating reaction on the servos displacing the joints.

ACKNOWLEDGEMENTS

The research of M. Karásek was supported by FRIA fellowship from F.R.S.-FNRS in Belgium (FC 89554).

REFERENCES

- [1] G. K. Taylor and A. L. R. Thomas. Dynamic flight stability in the desert locust *schistocerca gregaria*. *Journal of Experimental Biology*, 206(16):2803–2829, Aug. 2003.
- [2] M. Sun and Y. Xiong. Dynamic flight stability of a hovering bumblebee. *Journal of Experimental Biology*, 208:447–459, 2005.
- [3] C. T. Orłowski and A. R. Girard. Dynamics, stability, and control analyses of flapping wing micro-air vehicles. *Progress in Aerospace Sciences*, 51:18–30, 2012.
- [4] M. Karasek and A. Preumont. Flapping flight stability in hover: A comparison of various aerodynamic models. *International Journal of Micro Air Vehicles*, 4(3):203–226, 2012.
- [5] C. P. Ellington. The novel aerodynamics of insect flight: application to micro-air vehicles. *Journal of Experimental Biology*, 202:3439–3448, 1999.
- [6] L. Ristroph, G. Ristroph, S. Morozova, A. J. Bergou, S. Chang, J. Guckenheimer, Z. J. Wang, and I. Cohen. Active and passive stabilization of body pitch in insect flight. *Journal of The Royal Society Interface*, 10(85):1–13, 2013.

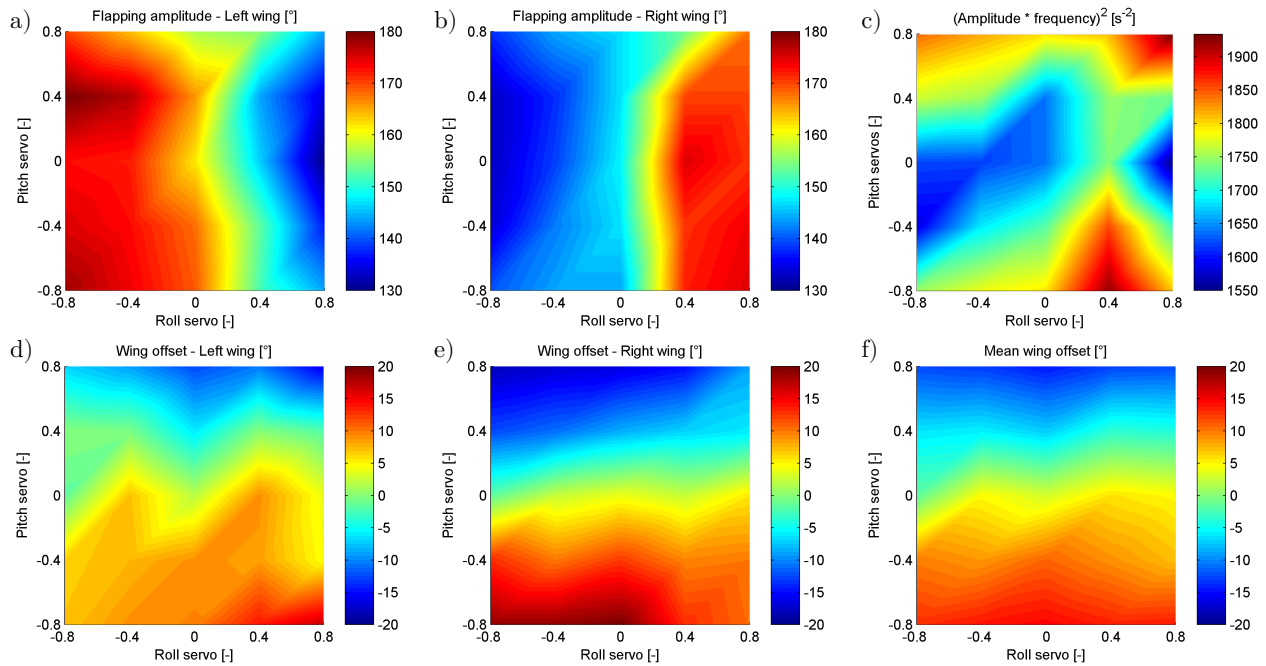


Figure 17: Kinematics measurements for combined pitch and roll servo commands: a) left amplitude, b) right amplitude, c) square of mean amplitude times frequency, d) left offset, e) right offset, f) mean offset.

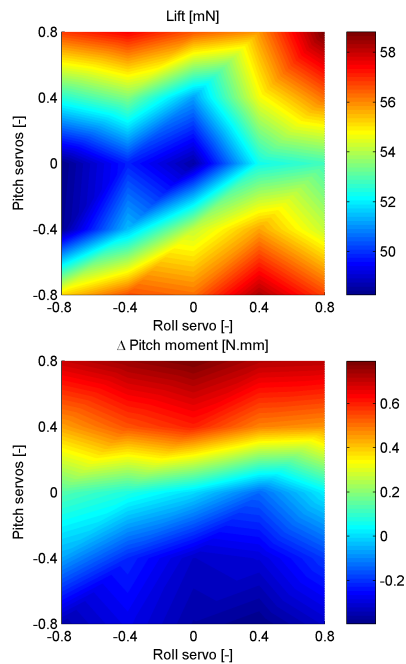


Figure 18: Force balance measurements for combined pitch and roll servo commands: lift (top) and pitch moment (bottom). Lift is tied to the square of mean amplitude times frequency (Figure 17 c) and moment corresponds to the mean offset (Figure 17 f).

- [7] G.C.H.E. de Croon, K.M.E. de Clerq, R. Ruijsink, B. Remes, and C. de Wagter. Design, aerodynamics, and vision-based control of the delfly. *International Journal of Micro Air Vehicles*, 1(2):71–97, Jun. 2009.
- [8] F. van Breugel, Z. E. Teoh, and H. Lipson. A passively stable hovering flapping micro-air vehicle. In D. Floreano, J.-C. Zufferey, M. V. Srinivasan, and C. P. Ellington, editors, *Flying Insects and Robots*, pages 171–184. Springer Berlin Heidelberg, 2010.
- [9] C. Richter and H. Lipson. Untethered hovering flapping flight of a 3d-printed mechanical insect. *Artificial life*, 17(2):73–86, 2011.
- [10] M. Keennon, K. Klingebiel, H. Won, and A. Andriukov. Development of the nano hummingbird: A tailless flapping wing micro air vehicle. *AIAA paper 2012-0588*, pages 1–24, 2012.
- [11] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood. Controlled flight of a biologically inspired, insect-scale robot. *Science*, 340:603–607, 2013.
- [12] M. Karasek, Y. Nan, I. Romanescu, and A. Preumont. Pitch moment generation and measurement in a robotic hummingbird. *International Journal of Micro Air Vehicles*, 5(4):299–309, 2013.

A Computational Study on the Aerodynamics of a 90 mm Ducted Contra-Rotating Lift Fan

Yangang Wang,* Weixiong Chen, Haitong Wang, and Ruiyu Li

School of Power and Energy, University of Northwestern Polytechnical University, Youyi WestStreet, Xi'an, P. R. China

ABSTRACT

In this paper, a steady 3D numerical study has been conducted on a 90 mm counter rotating lift fan to investigate its aerodynamic performance. The effect of axial distance between the fans and rotor speed are studied. A low thrust generation has been revealed from the simulation results due to the unfavourable working condition for the upstream rotor. Subsequently, an inlet guide vane (IGV) is geometrically designed to cause a pre-swirl effect at the inlet. Results show that the pressure ratio and total thrust are significantly improved at 10% by the IGV with a little sacrificing in efficiency. Such work can be a guidance in the further design and optimization.

Keywords: Contra-rotating lift fan; Rotation Speed; Axial Space; Pre-swirl

Nomenclature

C	blade chord length at hub
T_0	inlet total temperature
P_0	inlet total pressure
q_m	mass
β	flow angle
β_1	inlet flow angle
β_2	outlet flow angle
β_{1k}	inlet metal angle
β_{2k}	outlet metal angle
c	absolute flow velocity
u	rotation velocity
w	relative flow velocity
θ	rotation velocity
π_k^*	total pressure ratio
α	attack angle

1 INTRODUCTION

Micro air vehicles (MAVs) are flying vehicles with a proposed small geometric dimensions, which has been fostered continuous research due to their potential applications in civilian and military. Existing MAVs can be briefly categorized into three different concepts: fixed, rotary and flapping wing. Fixed and out-rotary concepts has drawn a tremendous research in the past decades, which benefits from the conventional technologies. However, they have been observed to

suffer from an unfavorable aerodynamics in minimizing the dimensional. Flapping wing MAVs, as the most intriguing type in MAV family are regarded as the most efficient prototype, while their difficulties in control and maneuver restrict its develop. Ducted fan MAV, as one special rotary MAV has obtained huge attention recently. The well documented advantage of the ducted fan MAV is its efficiency due to duct lip and enhanced blade tip lift. Moreover, the shrouded blades make such an MAV able to achieve a near wall mission and low noise production. The ducted fan submersed in the MAVs are normally driven by propellers or lift fans to perform a vertical/short takeoff and landing (VSTOL) capability. Instead of a single lift fan, the ducted fan aircrafts are normally equipped with contra-rotating fans [1, 2] in order to increase the net thrust and minimize the contra rotating torque which is suitable for a tailless configuration.

In view of the aerodynamics, the rear rotating fan refines the wake direction closer to the axial flow, and then enhance the performance [3]. The flow field of a contra-rotating ducted fan is extremely complex due to mutual aerodynamic interference effects of the high-speedy rotating fans. The aerodynamic of the rotors can significantly influence the overall performance of ducted fan MAVs. Relative studies are well documented in literatures. Guerrero et al.[4] proposed an aerodynamic prediction methodology that can be used for conceptual designing ducted-fan vertical takeoff and landing unmanned air vehicle. Their method is based on a parametric geometry model and semi-empirical aerodynamic methods, and the results were validated with wind tunnel test data on a full-scale vehicle. Williams et al.[5] performed the steady and unsteady aerodynamic studies on ducted fans using a frequency domain panel method based on three-dimensional linear compressible lifting surface theory. Their result shows that the duct has an adverse effect on the aero-elastic stability of the rotor. Page[6] developed a vortex blade-element propeller design and analysis procedure to the design and analysis of single- and counter-rotation ducted fans, which can be used on a single-rotation ducted fan with a fixed stator or a dual-rotation ducted fan. Stiltner et al.[7] performed a detailed design phase of a ducted fan MAV. With the detailed design phase completed, they moved forward with the construction of the MAV and improved some ways to fabricate the vehicle. Finally, tests including thrust testing, landing gear drop tests, electronic benchmarking and the beginning of tethered tests on the duct fan MAV were conducted on the constructed vehicle. Sharma[8, 9] investigated the influence

*Email address(es): wyg704@nwpu.edu.cn

of axial gap on the performance of upstream rotor combining a low-speed (maximum speed 2,500 r / min) test rig, their studies indicate that the axial gap between the rotors has an important impact on the performance, and thus affect the performance of the compressor. At a small axial clearance, the rear rotor can effectively delay the stall on the front rotor, however, they didn't make a detailed explanation of the flow mechanism. Later on, Wang et al.[10] continuously studied the effect of axial space between two blade rows, they indicated that the axial space can significantly affect the performance of fan, and it is always essential to optimize a critical space by trading off with the structure limitation.

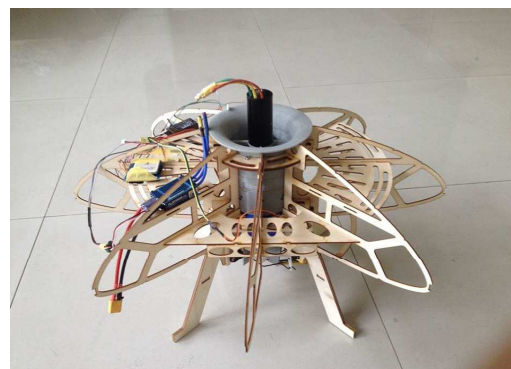
Since 2013, our group from the School of Power and Energy in the Northwestern Polytechnical University has engaged in the research of ducted fan MAVs regarding the related technologies in aerodynamics, mechanism, electronics and autonomy, etc. A remote controlled ducted MAV (named as FanSO) driven by ducted lift fan has been developed recently, see in Figure 1(a). The MAV is fabricated by balsa wood and Polylactide (PLA). It is 590 mm in length and 380 mm in height. Two batteries with the capacity of 5000mAh each can perform a 10 min flight duration. Figure 2(b) shows the details of the contra-rotating ducted lift fan equipped on the FanSO. The diameter of the hub and shroud is 36.4 mm and 90 mm, respectively. It is around 320 mm long. Two 12-blades fans are parallel mounted along the axis with a axial space around 22 mm. At working speed (RPM=26000), the lift fan can achieve maximal 25 Newton net lift and 3.5 kW power. Our purpose to obtain a deep understanding into the ducted fan flyer has triggered both experimental and computational studies. The experiences gained from the research activities will be used for further development and optimization. It is, however, realized from the free flight test that the commercially ordered lift fan cannot achieve an expected lift which inspires us to perform a study into the inner flow behavior.

In this paper, computation is conducted solely on the ducted lift fan isolated from the MAV to predict the aerodynamic characteristics of the lift fan. Effect of axial space, inlet/outlet metal angle and RPM are evaluated in terms of thrust production, energy consumption and efficiency. In addition, an inlet guide vane (IGV) is developed and introduced into the lift fan system which provides a considerable improvement in the thrust generation due to the pre-swirl effect. The present study will guide the further design and optimize of the ducted fan MAV.

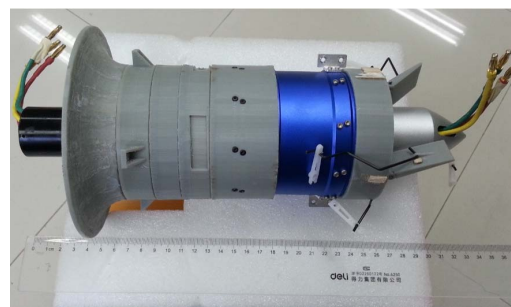
2 METHODOLOGY

The geometry of the fan blade is obtained by a 3D Laser scanning machine, which can resolve down to 0.02 mm. Figure 2 geometrically shows the schematics of the lift fan.

The commercial software NUMECA Autogrid was employed to generate the grid system of the lift fan. The blade and hub surface is meshed with Cartesian O and H mesh



(a) NPU-1



(b) Ducted Contra-Rotating Fan

Figure 1: NPU-1 and the ducted contra-rotating lift fan

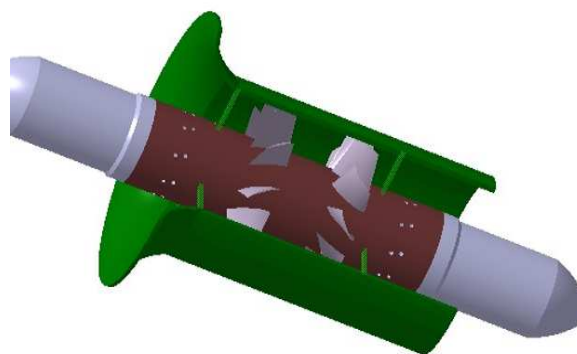


Figure 2: 3D topology of the ducted lift fan rendered in CATIA.

while the 3D space is filled with O4H type mesh, as shown in Figure 3. The butterfly grid was used for the tip clearance between the blade tip and the shroud. A refined y^+ is set less than 2.0 at the nonslip and adiabatic wall boundary conditions for capture the boundary layer flow phenomena which results in 1.1 million cells in total. Note that only one passage is

meshed and computed, the entire flow field will be mirrored later on.

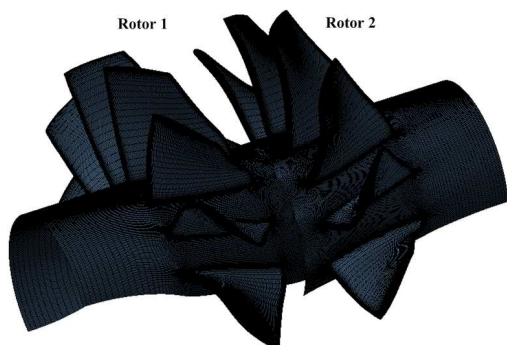


Figure 3: Grid system of the lift fan.

The steady simulation for the counter-rotating lift fan were performed using the flow solver EURANUS. It computed the conservative Reynolds Averaged Navier-Stokes (RANS) equations coupled with one equation Spalart-Allmaras (S-A) turbulent model which is widely used to investigate the effect of viscosity in turbomachinery. In our case, the Reynolds number of the counter-rotating lift fan is about half million. In the work of this paper, the mixing plane between two rows was treated using conservative coupling by pitch-wise rows approach, which was able to guarantee strict conservation of mass, momentum and energy through the interface of the two blade rows. The controlling equations are spatially discretized via a cell-centred finite volume formulation, for steady simulations, the equations were solved using an explicit four-order Runge-Kutta method with a local time step to obtain steady-state solution. The computational cost can be dramatically reduced by local time marching, implicit residual smoothing, and multi-grid techniques. The flow angle β , total pressure (P_{01}), and total temperature (T_{01}) are specified to be uniform across the whole inlet area in accordance to actual operation point, chock point. That means the inlet and outlet area were set as the local atmosphere condition ($P_{01}=P_2=101325$ Pa, $T_{01}=288.15$ K,). The performance curves were achieved by increasing the rotating speed at the chock point of the lift fan.

3 AERODYNAMIC PERFORMANCE

3.1 Effect of Axial Space

Figure 4 depicts the influence of axial spacing on mass, pressure ratio, efficiency, thrust, power at design speed at 11 different axial spaces. It can be seen that the pressure ratio, axial thrust and power follow with an asymptotical likelihood tendency and almost remain the same with the increasing axial space. The rotor1 efficiency almost remain the same with the increasing axial space while the rotor2 and total ef-

iciency increased firstly and then decline along with the increasing axial space, where the maximal value (83.5% and 80%, respectively) peaks at 0.7c. It can be conjectured that at small axial space (say below 0.5), the upstream rotor wake at the exit cannot fully mixed with the mainstream, which results in an unfavorable incoming flow for the downstream rotor. When increase the axial space (say between 0.5-0.7), upstream wakes and mainstream can be fully mixed during the relative long distance, which provide an better incoming flow for the downstream rotor, hence, gives a higher efficiency ; Continuing increasing the axial space, mixing loss will predominate the flowfield, which leads to a decline of the efficiency.

3.2 Flow Angle Analysis

As known, the effective angle of attack can be derived from the difference between the inlet flow and metal angle, say $\alpha = \beta - \beta_k$. The preferable working condition happens at a positive angle of attack. Figure 5 plots the inlet/outlet flow and metal angles along the blade height for both upstream and downstream rotors. It can be seen from Figure 5(a) that the upstream rotor angle of attack turns lower gradually along the increase of blade height, and the AOA turns to negative when the radius is more than about 30% of the blade span. It means that the load magnitude of rotor decrease gradually along the blade height, which can explain the low total pressure at the upstream rotor. Meanwhile, negative AOA will bring greater pressure loss due to boundary layer on pressure surface, which can explain the low efficiency at the upstream rotor. The angle distribution of the downstream rotor performs a similar behavior to the upstream rotor which can be revealed from Figure 5(b). However, due to contra-rotating effect, the downstream rotor works at a positive angle before 65% blade span, and the difference is very small even though the AOA after 65% blade span is negative, which results in a larger turning angle of airflow and improves the working capacity of the downstream rotor. The loss level is also relatively less than the upstream rotor, so as to retain high efficiency and gain greater thrust. In addition, by looking into Figure 5, the blades are mostly working at high angle of attack either in positive or negative circumstance. Some unfavorable aerodynamic phenomena such as decambering and separation might happen at such working conditions, which inspired us to optimize the ducted fan configuration. One way to improve the blade performance is to generate a pre-whirled flow at the inlet for changing the flow angle. A design and optimization of an inlet guide vane (IGV) will be subsequently discussed in the next chapter.

4 PRE-SWIRL AT INLET

4.1 Design Method of IGV

Figure 6 schematically illustrates the change of the inlet flow angle when the ducted fan are mounted with a pre-swirled IGV. It can obviously note that the rotor can work at

a positive attack angle with IGV.

A six-blade IGV is designed by multiple circular arc method, see in Figure 7. The projected meridional chord is 20 mm, the leading and trailing edge radius are 0.25 mm and 0.15 mm, respectively. The maximal thickness of the IGV blade is 1.5 mm and 15 mm away from the upstream rotor.

4.2 Aerodynamic Performance Comparison

The same numerical methodology as above has been applied on the ducted fan with a equipped IGV to predict its aerodynamics. Table 1 shows comparative performance parameters between the original lift fan model (M0) and modified model with IGV (M1) at design speed, it can be seen that the pre-whirl effect by the IGV increases the upstream rotor load and the total pressure ratio. For the upstream rotor, the efficiency and the thrust increased by 7% and 168%, respectively. Due to change in upstream rotor load, the flow conditions of downstream rotor changes as well, which leads to a slightly decline of pressure ratio, efficiency and thrust; in addition, the inlet guide vane creates negative thrust by 2.579 N. However, the pre-swirl effect caused by the IGV can enhance the aerodynamic performance of the lift fan. The entire thrust increased by about 10%, while the efficiency decreases by about 1.6% due to the low efficiency of the downstream rotor and aerodynamic loss of IGV.

4.3 Inlet/Outlet Flow Angle Analysis

A detailed inlet/outlet flow angle between the original lift fan and the modified one with IGV is shown in Figure 8. Figure 8(a) shows that the inlet flow angle increases around 10° along with the spanwise position. At this moment, the majority of the blade working at a positive AOA which can product more lift corresponding to the results listed in Table 1. However, a negative effect caused by the IGV can be found in Figure 8(b) that the attack angle decreases compared with the original lift fan configuration which will decrease the aerodynamic performance of the downstream rotor.

In order to obtain a detailed understanding of the flow behavior inside the ducted fan, the static pressure contour on the blade surface as well as the IGV are plotted in Figure 9(a). It can be seen that the upstream rotor works at positive attack angle conditions due to the deceleration, pressure diffusion and stream guidance of IGV. Proper flow conditions were achieved for the upstream rotor, i.e. freedom from big separation means lower flow losses and correspondingly higher efficiency. The AOA and hence the blade load of R2 has declined as a result of the changed flow condition from R1, therefore, lower efficiency was observed for the fact that the proportion of frictional loss has increased ultimately.

The total pressure distribution was obtained at four different axial locations on S3 planes as in Figure 9(b). The total pressure ratio remains at the same level for the two rows of the rotor, which is beneficial for the reduction of gyroscopic torque. However, the boundary layers of both R1 and R2 are thick which can be seen on the S3 planes located right down-

stream of blades, therefore an increment of wake width and strength is to be expected as well as lower efficiency. In order to save the cost of design and manufacture, raw materials of PLA were used although it is weak in rigidity and strength. What becomes apparent is that the thickness of blades has to be increased, so do the profile losses and the flow drag, which must be paid much attention in further study.

4.4 Effect of Rotary Speed

Figure 10 shows the Speed characteristics comparison between the original lift fan and the modified lift fan with IGV. It can be found that: under different speed, the pressure ratio, axial thrust and power increase as the rotating speed increases both in original lift fan and the modified one. The total efficiency increases firstly and then decline along with the increasing rotating speed, where the maximum value peaks at 26000 rpm for the original one, and for the modified one, it is 22000 rpm that the efficiency achieve optimal. But for the original fan, the thrust or pressure ratio of rotor1 is about 1/3 of rotor2, which means the load of rotor1 is relatively smaller than expected. The thrust contribution of rotor1 is approximately equal to rotor2 for the modified lift fan, so it is with the pressure ratio, efficiency and power. In consideration of our goal of increasing the total thrust of the ducted fan, which determines whether the MAV could fly. It is beneficial for the contra-rotating fan to develop its performance better though the overall efficiency reduce a little after adding a row of IGV.

5 CONCLUSIONS

The paper conducted a numerical research on a 90mm diameter contra-rotating lift fan. The effect of axial space and speed are investigated. Due to the low thrust generated by this commercial ducted fan, an IGV has been parametrically designed to achieve a better thrust production which can guide our further design. The main conclusions are listed below:

(1) The axial space of the two blade rows significantly affect the overall aerodynamic performance. The mass flux, pressure ratio and axial thrust almost remain the same with the increasing axial space, while the efficiency firstly increases till $0.7c$ and then decreases.

(2) The reason of rotor1s poor working capacity has been revealed by analyzing the inlet/outlet and metal angles distribution along with the spanwise. A pre-swirl effect caused by rotor1 can enhance the aerodynamic performance.

(3) Equipping an IGV at the inlet of rotor1 makes a significant effect on the flow field, the efficiency of rotor1 increases by about 7%, and thrust increased by nearly 168%; even though the IGV creates a negative thrust of 2.759N, the total thrust increases by about 10%.

(4) The pressure ratio, axial thrust and power of both configurations (with/without IGV) increase as the increasing rotating. The total efficiency increases firstly and then declines along with the increasing rotating speed, where the maximum value peaks at 26000 rpm for the original one, and for the

Table 1: Performance Parameters Comparison of Original and Modified Design.

	$q_m(kg/s)$		π_k^*		Efficiency		Axial Thrust(N)		Power(kW)	
	M0	M1	M0	M1	M0	M1	M0	M1	M0	M1
IGV	\	\	\	0.993	\	\	\	-2.759	\	\
Rotor1	\	\	1.016	1.032	0.732	0.808	6.122	16.43	1.003	1.876
Rotor2	\	\	1.038	1.029	0.832	0.801	19.75	14.77	2.089	1.747
Total	0.554	0.581	1.055	1.060	0.797	0.781	25.87	28.44	3.091	3.623

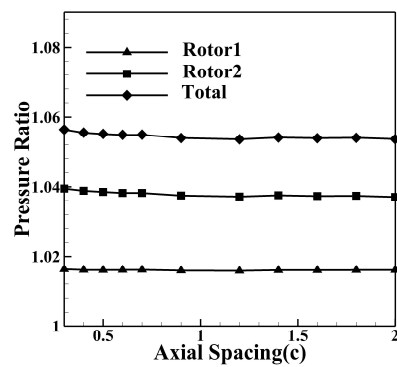
modified one, it is 22000 rpm that the efficiency achieve optimal. The thrust contribution of rotor1 is approximately equal to rotor2 for the modified lift fan after adding an IGV, that means the IGV is beneficial for the contra-rotating fan to develop its performance better though the overall efficiency reduce a little.

ACKNOWLEDGEMENTS

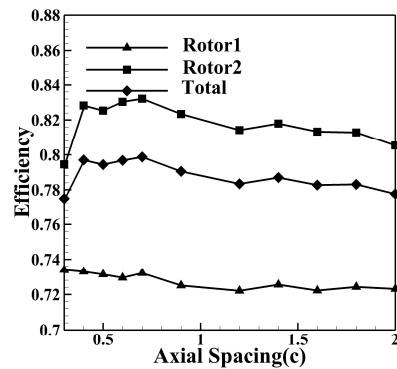
The technical assistance from the NPU turbomachinery laboratory in providing the scanning geometry data is of great appreciated. This work has been financially supported by the National Natural Science Foundation of China [grant number 51376150]. All these supports are gratefully acknowledged.

REFERENCES

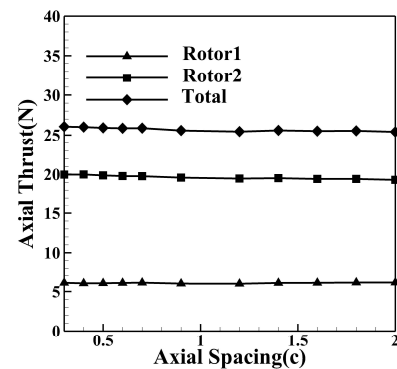
- [1] T.G. Sylvester, R.J. Brown, and C.F. O'Connor. F-35b lift fan inlet development. In *AIAA Centennial of Naval Aviation Forum "100 Years of Achievement and Progress"*, 21-22, September 2011.
- [2] G.P. Walker and D.A. Allen. X-35b stovl flight control law design and flying qualities. In *Biennial International Powered Lift Conference and Exhibit*, 5-7, November 2002.
- [3] R.R. Harrington. *Full-Scale-Tunnel Investigation of the Static-Thrust Performance of a Coaxial Helicopter Rotor*. NACA TN-2318, New York, NY, 1951.
- [4] I. Guerrero, K. Londenber, P. Gelhausen, and A. Myklebust. A powered lift aerodynamic analysis for the design of ducted fan uavs. In *nd AIAA "Unmanned Unlimited" Systems, Technologies, and Operations*, 15-18, September 2003.
- [5] M.H. Williams, J. Cho, and W.N. Dalton. Unsteady aerodynamic analysis of ducted fans. *Journal of Propulsion*, 7(5):800–804, 1991.
- [6] G.S. Page. Design and analysis of single and dual rotation ducted fans. In *AIAA 34th Aerospace Sciences Meeting and Exhibit*, 15-18, Jan 1996.
- [7] B.C. Stiltner et al. Detailed design, construction, and flight tests of a remotely piloted ducted fan mav. In *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 4-7, Jan 2010.
- [8] P. B. Sharama, Y. P. Jian, and D.S. Pundhir. Study of some factors affecting the performance of a contra-rotating axial compressor state. *Proceedings of the Institution of Mechanical Engineers, Part A: Power and Process Engineering*, 15(211), 1985.
- [9] P. B. Sharama, Y. P. Jian, and D.S. Pundhir. Stalling behavior of a contra-rotation axial compressor stage. *IS-ABE*, 85(7087), 1985.
- [10] Y. Wang, N. Niu, and B. Liu. two stage axial spacing of counter rotating compressor impact on performance. *Journal of Aerospace Power*, 25(3), 2010.



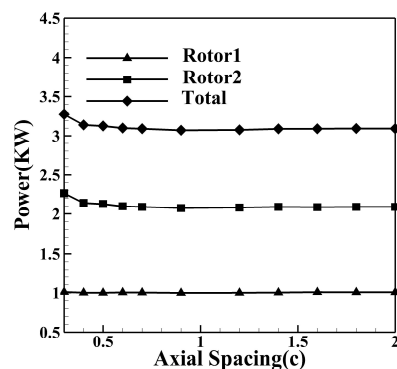
(a) Pressure ratio vs. axial space



(b) Efficiency vs. axial space

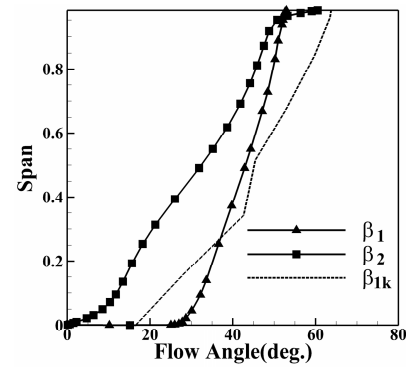


(c) Axial thrust vs. axial space

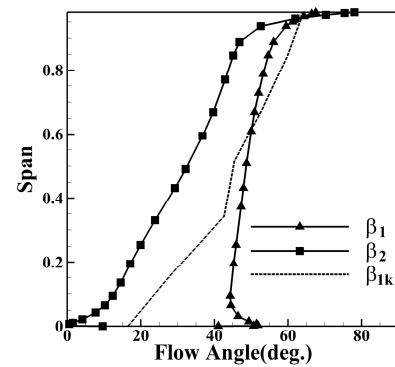


(d) Power vs. axial space

Figure 4: Aerodynamic performance of the axial space effect.



(a) Upstream rotor



(b) Downstream rotor

Figure 5: The spanwise flow angle distribution.

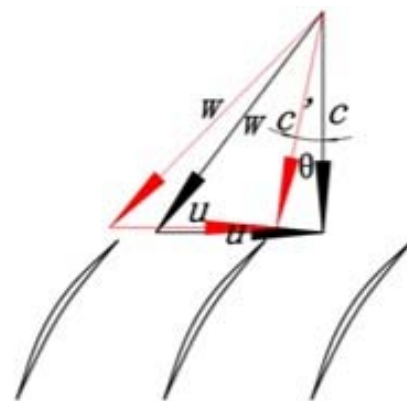


Figure 6: Velocity triangle comparison of rotor 1 at 50% span.

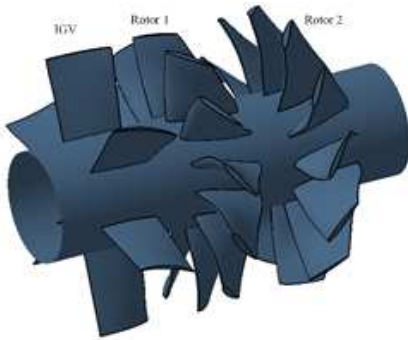
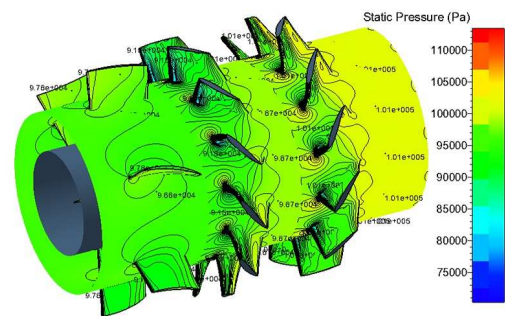
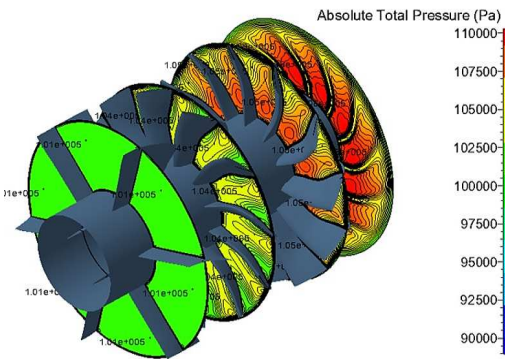


Figure 7: 3D configuration of the IGV mounted on the lift fan.

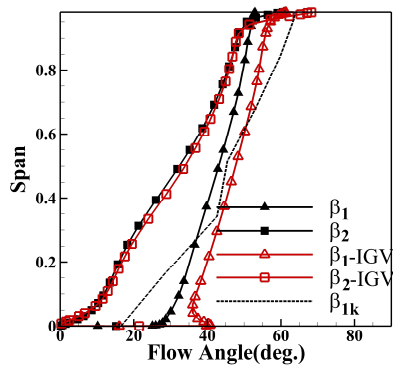


(a) Static pressure distribution

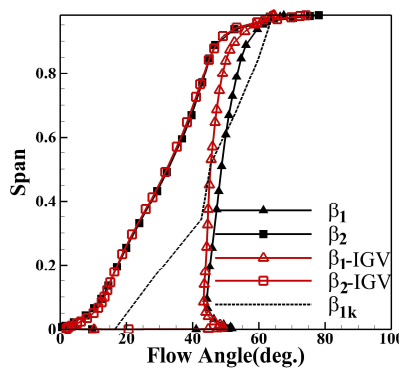


(b) Total pressure distribution on S3 planes

Figure 9: Pressure contour.

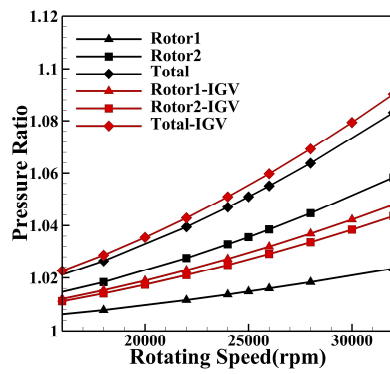


(a) Upstream rotor

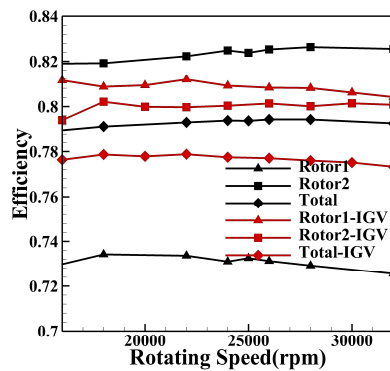


(b) Downstream rotor

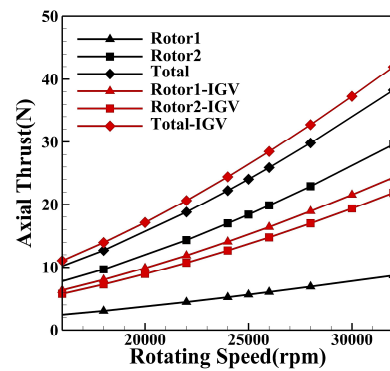
Figure 8: The spanwise flow angle distribution.



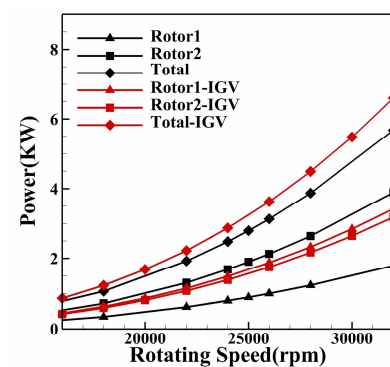
(a) Pressure ratio vs. RPM



(b) Efficiency vs. RPM



(c) Axial thrust vs. RPM



(d) Power vs. RPM

Figure 10: Speed characteristics comparison between original lift fan and modified lift fan with IGV.

Towards Estimation and Correction of Wind Effects on a Quadrotor UAV

Fabrizio Schiano*, Javier Alonso-Mora[†], Konrad Rudin[‡], Paul Beardsley[§], Roland Siegwart[¶], Bruno Siciliano^{||}

ABSTRACT

In this paper the problem of estimation and correction of wind effects on a quadrotor UAV, without using wind sensors, is discussed. A large body of research addresses the effects of wind on a quadrotor, however most of them consider it only as a disturbance in their control loop and, for this reason, they solved the problem compensating for the wind with a powerful controller. The main part of this paper is related to the modeling of wind on a quadrotor and to the wind tunnel tests performed at the IFD wind tunnel of ETH Zurich. The approach presented can be used as a starting point for future works. The results obtained in the wind tunnel are really promising for the formulation of a complete aerodynamic model of the quadrotor that has been missing until now.

1 INTRODUCTION

Quadrotors are becoming really popular in recent years both for their field of application and, for the multidisciplinary applications that they offer (i.e. Mechanics, Electronics, Control Theory, Aerodynamics and so on). They are useful for their high capability of Vertical Take Off and Landing (VTOL), high controllability and maneuverability. Moreover, unlike most fixed wing UAVs, the quadrotors are able to hover in a certain position. For all these reasons, they are more suitable to be used for tasks such as of mapping and 3D- image-reconstruction of a specific area. Moreover, they also provide

good capabilities in building construction [1], monitoring of public places, avoidance of terroristic attacks, in search and rescue missions after an environmental disaster and entertainment [2].

The goal of this work is to estimate and correct wind effects on a quadrotor [3] without adding wind sensors on it (because of the high cost and difficulty to install them on a quadrotor).

In most of the cases reported in literature, wind is not considered explicitly, and so, the controller uses it just as a disturbance to reject. In some cases it would be useful to estimate wind in order to have a wind measurement to add to the control loop and to save money for buying a wind sensor or to have a *quadrotor anemometer* wherever needed.

This paper is organized as follows. First, the model of the quadrotor, focusing on the role of wind into the equations of the model, will be analyzed. Then, a short overview of the platform used for the real experiments is presented. The control strategy used to control the quadrotor will then be presented and finally, the approach to estimate wind giving a main focus to the tests performed in the wind tunnel to identify some of the parameters belonging to the aerodynamic model of the quadrotor will be introduced.

2 MODEL OF THE QUADROTOR

The main equations of the model of the quadrotor come from [4] and [5]. The starting point for their formulation is the application of the Newton-Euler approach¹ to rigid body dynamics equations. Furthermore, the modeling assumptions are reported below:

- 1 The input of the system is the speed of the rotors.
- 2 The outputs of the system are its position and attitude.
- 3 The CoG and the body frame origin are assumed to coincide.
- 4 The propellers are supposed to be rigid.

The rigid body dynamics equations can be split into two parts: force equations (Equation 1) and moment equations (Equation 2).

$$m\mathbf{I}_{3 \times 3}\dot{\mathbf{v}} + \boldsymbol{\omega} \times m\mathbf{v} = \mathbf{F} \quad (1)$$

¹The Newton-Euler approach is based on the equilibrium of forces and moments acting on the object analysis.

*Fabrizio Schiano received both his B.Sc and M.Sc in Automation Engineering from the University of Naples Federico II. He is a researcher at Zentrum für Telematik (Gerbrunn, Germany) since July 2013. fabrizio.schiano@gmail.com

[†]Javier Alonso-Mora is a researcher at the Autonomous Systems Lab at ETH Zurich and Disney Research Zurich jalonso@disneyresearch.com

[‡]Konrad Rudin is a Ph.D candidate at the Autonomous Systems Lab at ETH Zurich konrad.rudin@mavt.ethz.ch

[§]Paul Beardsley is Principal Research Scientist at Disney Research Zurich. pab@disneyresearch.com

[¶]Roland Siegwart is full professor for autonomous systems at ETH Zurich and Director of the Autonomous Systems Lab at ETH Zurich. rsiegwart@ethz.ch

^{||}Bruno Siciliano is full professor of Control and Robotics, and Director of the PRISMA Lab at University of Naples Federico II. siciliano@unina.it

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} = \boldsymbol{\tau} \quad (2)$$

With:

m : Body mass

$\mathbf{I}_{3 \times 3}$: Identity Matrix

$\boldsymbol{\omega}$: Body angular speed

$\mathbf{F}$: External force

$\mathbf{J} \in \mathbb{R}^{3 \times 3}$: The Inertia matrix

$\mathbf{v}$: Body linear speed

$\boldsymbol{\tau}$: External moment

These terms have the following expression:

$$\mathbf{I}_{3 \times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad (3)$$

$$\mathbf{J} = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (4)$$

Particular attention was given to the inertia matrix $\mathbf{J}$ and for this reason a CAD model of the quadrotor, shown in Figure 1, was designed.

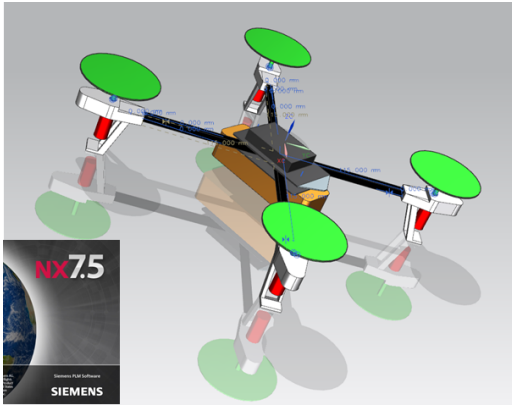


Figure 1: CAD Model of the Quadrotor

$$\mathbf{J} = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} \quad (5)$$

The results from the CAD software made clear that the products of inertia can be neglected. As expected, the terms on the diagonal are different from each other because the real quadrotor is not symmetric with respect to x, y and z . Therefore, the inertia matrix used in the model is:

$$\mathbf{J} = \begin{bmatrix} 1.79e-3 & 0 & 0 \\ 0 & 1.95e-3 & 0 \\ 0 & 0 & 3.56e-3 \end{bmatrix} [Kg \cdot m^2] \quad (6)$$

Equation 1 and 2 can be written in a compact form:

$$\begin{bmatrix} m \mathbf{I}_{3 \times 3} & 0 \\ 0 & \mathbf{J} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega} \times m \mathbf{v} \\ \boldsymbol{\omega} \times \mathbf{J} \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \boldsymbol{\tau} \end{bmatrix} \quad (7)$$

Introducing the *Euler angles*:

$$[\phi \ \theta \ \psi]^T \quad (8)$$

their time derivative will be related to the angular rates of the quadrotor in the inertial frame through the following expression:

$$[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T = \mathbf{M}^{-1} [\omega_{x_i} \ \omega_{y_i} \ \omega_{z_i}]^T \quad (9)$$

where the $\mathbf{M}$ matrix is:

$$\mathbf{M} = \begin{bmatrix} \frac{c\psi}{c\theta} & \frac{s\psi}{c\theta} & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

In order to have the relation between the time derivative of the Euler angles and the angular rates of the quadrotor in the body frame the following equation was used:

$$[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T = \mathbf{M}^{-1} [\omega_{x_b} \ \omega_{y_b} \ \omega_{z_b}]^T \quad (11)$$

where the $\mathbf{R}_B^I$ matrix is the rotation matrix which expresses the rotation to go from the body frame to the inertial frame:

$$\mathbf{R}_B^I = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (12)$$

where $c\psi = \cos(\psi)$, $s\psi = \sin(\psi)$.

In the same way, it is possible to get the linear velocities in the body frame from the linear velocities in the inertial frame:

$$[\dot{x}_b \ \dot{y}_b \ \dot{z}_b]^T = (\mathbf{R}_B^I)^{-1} [\dot{x}_i \ \dot{y}_i \ \dot{z}_i]^T \quad (13)$$

The next step consists of the characterization of Equation 7 for the quadrotor. It is easier to look separately at the force and moment equations.

The model of the quadrotor was obtained by considering that it is near to the hover condition. In this case, the main forces and moments come from the propellers. Moreover, from Equation 10 and 12, it is clear that the $\mathbf{M}$ and $\mathbf{R}_B^I$ matrices become:

$$\mathbf{M}|_{(\phi, \theta, \psi=0)} = \mathbf{R}_B^I|_{(\phi, \theta, \psi=0)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (14)$$

$$\Rightarrow [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T = [\omega_{x_i} \ \omega_{y_i} \ \omega_{z_i}]^T = [\omega_{x_b} \ \omega_{y_b} \ \omega_{z_b}]^T \quad (15)$$

2.1 Force Equations

The thrust generated by a propeller is proportional to the square of its rotational speed [6]. For the purposes of this work the thrust generated by a propeller will be addressed just as a term proportional to the square of its rotational speed:

$$T_i = b \Omega_i^2; \quad b : \text{thrust constant} \quad (16)$$

The equilibrium of the forces can be written as follows:

$$\begin{bmatrix} \ddot{x}_i \\ \ddot{y}_i \\ \ddot{z}_i \end{bmatrix} = - \begin{bmatrix} \omega_{x_b} \\ \omega_{y_b} \\ \omega_{z_b} \end{bmatrix} \times \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} + g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{\mathbf{F}_{wI}}{m} - \frac{T_1 + T_2 + T_3 + T_4}{m} \mathbf{R}_B^I \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (17)$$

where $\mathbf{F}_{wI}$ represents the disturbance force due to uncontrollable factors such as wind. It is important to highlight that this term represents force disturbances related to wind only under the assumption that other force disturbances can be neglected.

2.2 Moment Equations

The drag moment generated by a propeller is proportional to the square of its rotational speed [6]. The thrust and the drag moment generated by a propeller will be considered just as a term proportional to the square of its rotational speed:

$$D_i = d \Omega_i^2; \quad d : \text{drag constant} \quad (18)$$

The equilibrium of the moments can be written as follows:

$$\begin{bmatrix} \dot{\omega}_{x_b} \\ \dot{\omega}_{y_b} \\ \dot{\omega}_{z_b} \end{bmatrix} = -\mathbf{J}^{-1} \boldsymbol{\omega} \times \mathbf{J} \begin{bmatrix} \omega_{x_b} \\ \omega_{y_b} \\ \omega_{z_b} \end{bmatrix} + \mathbf{J}^{-1} \begin{bmatrix} L(T_4 - T_2) \\ L(T_1 - T_3) \\ D_1 - D_2 + D_3 - D_4 \end{bmatrix} - \mathbf{J}^{-1} \boldsymbol{\tau}_d \quad (19)$$

where $\boldsymbol{\tau}_d$ represents the disturbance torque due to uncontrollable factors such as wind. As for the force equations, this term represents moment disturbances related to wind only under the assumption that other moment disturbances can be neglected. For convenience, the following effects were neglected:

- Gyroscopic moments caused by the combination of rotations of the four propellers and vehicle frame.
- Friction torque due to rotational motion [7].

3 CONTROLLER

3.1 LQR Controller

The controller used for the quadrotor is shown in Figure 2 can be split into a *Position Controller* (off-board) and an *Attitude Controller* (on-board).

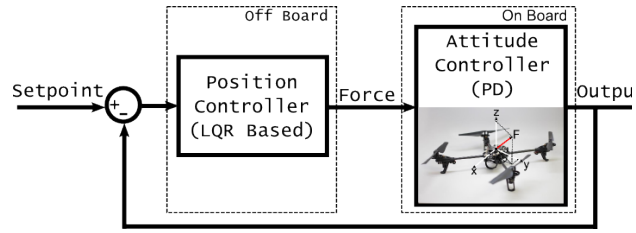


Figure 2: Scheme of the overall control structure

The *Position Controller* is a Linear Quadratic Regulator (LQR) giving the quadrotor a force vector that is interpreted on-board.

The *Attitude Controller* is a PD controller running at 250 Hz. The LQR control law is based on an algorithm which minimizes a cost function containing weighting factors chosen ad-hoc. It consists of an automated way to find the optimal state-feedback controller. The LQR controller is not the main goal of this work and, for this reason, only a short overview of it is provided. The goal of the LQR is to find the feedback control law $\mathbf{u}^*(t)$ which minimizes the quadratic cost function:

$$\mathbf{J}(\mathbf{x}(t), \mathbf{u}(t)) = \int_0^\infty \mathbf{x}(t)^T \mathbf{Q} \mathbf{x}(t) + \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t) dt \quad (20)$$

This is represented by:

$$\mathbf{u}^*(t) = -\mathbf{K}_{LQR} \mathbf{x}(t) \quad (21)$$

where $\mathbf{K}_{LQR}$ is given by:

$$\mathbf{K}_{LQR} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}(t) \quad (22)$$

where $\mathbf{P}$ is the solution of the continuous time *Algebraic Riccati Equation* (ARE) and $\mathbf{Q}, \mathbf{R}$ are:

$$\mathbf{Q} = \mathbf{Q}^T, \mathbf{Q} \geq 0; \mathbf{R} = \mathbf{R}^T, \mathbf{R} > 0. \quad (23)$$

It must be underlined that the LQR itself is not able to reject constant disturbances and, for this reason, in the following paragraph a solution to deal with this problem is presented.

3.2 LQRI Controller

One way to solve the problem previously mentioned is to add an integral part to the LQR controller implementing a so-called *LQRI* controller. The LQRI is based on the extension of the system's state vector with the integral of the output as shown in Figure 3. It is important to highlight why the constant disturbances are of interest. The two main reasons were:

- The mass of the quadrotor can change because different batteries, which have different weights, were used.
- The input given to the motors was an integer number $\delta \in [0 - 500]$ but, to link this δ to the thrust $\mathbf{T}$ generated

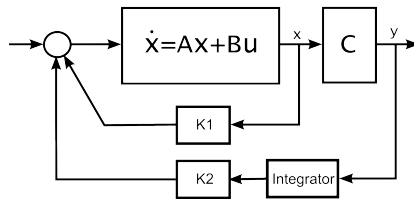


Figure 3: LQRI Controller Scheme

by the propellers, it was used a *thrust function* identified at the beginning of the PixHawk project [8]. It was noticed that the mapping between the δ and the T was wrong.

In control theory, in both cases described before, the result is a constant disturbance.

3.3 LQR-LQRI Results

In Figure 4-5, some results about the performances of the LQR and of the LQRI controller are showed. Both figures deal with the following test: in the interval 10–40s a setpoint at an altitude of 1 m was given to the quadrotor, followed by a setpoint of 1.5 m in the interval 40–60s, then again 1 m for a few seconds and finally the *land* command was given to the quadrotor.

From Figures 4-5 it can be observed that with the LQR controller there is a constant error of about 0.8 m, a big error which can not be accepted. An alternative to the LQRI controller could be to insert into the model of the quadrotor a fake mass which has to be changed every time there is a modification in the mass of the quadrotor, but this is not the best and the most formal solution. To solve this problem and to get rid of constant disturbances, the LQRI was implemented. Furthermore, several tests were performed to identify a better thrust function².

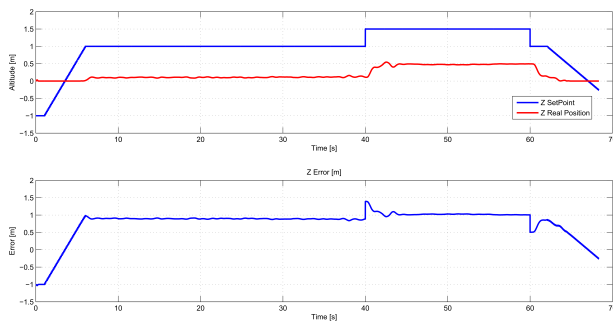


Figure 4: LQR Results in an experiment of different Altitudes

²The thrust function relates the input given to the motors to the effective thrust generated by them.

The improvements given from the LQRI are underlined in Figure 5. In the graph it can be observed that with the LQRI an error of about 0.015 m was achieved. It can be considered a good performance for the purposes of the study.

4 THE PLATFORM

The quadrotor used in this work is shown in Figure 6. It has the frame of the Parrot AR Drone 2.0 [9], with the same propellers, motors and motor controllers. The Parrot frame was selected for its low cost, good strength, and low weight (400g ca.). The main electronics related to the quadrotor are composed of:

- 1 PX4FMU Autopilot / Flight Management Unit.
- 2 PX4IOAR ARDrone Carrier/Adapter.
- 3 RF Laird Board Base Station (connected to the Real-Time PC).
- 4 RF Laird Board Module (connected to the quadrotor).

The quadrotor's power supply is represented by 3-cell LiPo batteries with 1800 mAh³ and a voltage of 11.1V.

The position and the attitude of the quadrotor were measured through the Vicon Motion Capture System. The main characteristics of this system of infrared cameras are:

- Speed: the system can be run at or below 375 Hz (in our experiments, a frequency of 200-250 Hz was used).
- Precision: experimental tests show that the deviations of position estimates for single static markers are of the order of 50 microns which is well beyond the requirements for a good flight.
- Robustness: using the VICON Tracker software, tracking of objects such as quadrotors is rarely lost, even during extreme situations such as fast maneuvers (speeds of 3.0m/s, accelerations of 12m/s² and angular speeds of 800 deg/s [10]).

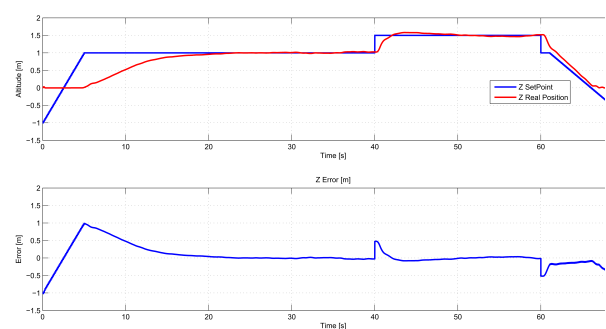


Figure 5: LQRI Results in an experiment of different Altitudes

³In this project both 1800 mAh and 2200 mAh batteries were used.



Figure 6: The quadrotor used in the tests with a focus on the VICON passive markers used to track it through the VICON system.

On the other hand, the software used for this work was mainly based on Matlab/Simulink[®] and the main feature of the MathWorks Software useful for the project was the *Real – TimeWindows TargetTM* [11]. This toolbox provides a real-time engine for executing Simulink models on a Microsoft Windows[®] PC and blocks to interface an application to a wide range of I/O boards. It allows to create and control a real-time system for rapid prototyping and hardware-in-the-loop simulation.

The communication link to and from the quadrotor was established thanks to RF communication through:

- 1 *RF Laird Board Base Station*: used as a TX/RX on the PC-Side (shown in Figure 7.a).
- 2 *RF Laird Module*: used as a TX/RX on the quadrotor-Side (shown in Figure 7.b).

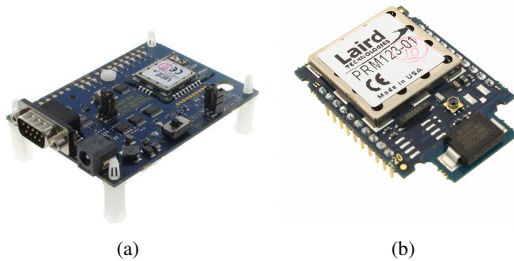


Figure 7: Communication: (a) RF Laird board Base Station and (b) RF Laird Module

5 WIND TUNNEL EXPERIMENTS

The main reasons to perform such experiments in the wind tunnel are listed below:

- 1 Identification of $c_D A$ coefficient, where c_D is the drag coefficient of the quadrotor and A is the area of the quadrotor interested by the flow of the wind [m^2].
- 2 Identification of $c_L A$ coefficient, where c_L is the lift coefficient of the quadrotor.

3 Identification of wind's influence on the magnitude of the Thrust Vector T .

The area A of the quadrotor interested by the flow of the wind is not easy to compute due to its irregular shape and for this reason in this work it was not separated respectively from the drag (c_D) and lift (c_L) coefficients. The identification performed, as previously explained, is of the whole terms $c_D A$ and $c_L A$.

The wind tunnel used, shown in Figure 8, is located at the facilities of the Institute of Fluid Dynamics (IFD) of ETH. In that department, there are three different wind tunnels and, for this work, the largest one was chosen.



Figure 8: Wind Tunnel of IFD-ETH (Image courtesy of IFD-ETH) [12]

To identify the coefficients listed before, it was needed to measure, during the experiments, the three forces acting on the quadrotor. For this purpose, a six-axis force balance was used (ATI Mini40-E Transducer in Figure 9.a).

In this Section, the different kinds of experiments performed in the wind tunnel will be described⁴. In particular, the experiments can be divided in three categories:

- 1 Yaw Experiments⁵.
- 2 Pitch Experiments⁶.
- 3 Thrust Experiments.

Both yaw and pitch experiments were performed with the propellers left loose, while the thrust experiments were performed changing the input to the motors.

The final result of the fixation of the quadrotor in the wind tunnel is showed in Figure 12 and a detailed view of the assembly between the force sensor and the two adapter plates is showed in Figure 9.b.

⁴The choice of such experiments is driven by time, money and the wind tunnel waiting list.

⁵The yaw experiments were performed at these angles: $[-90^\circ, -75^\circ, -60^\circ, -45^\circ, -30^\circ, -15^\circ, 0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ]$. The signs of the angles are chosen according to Figure 10.

⁶The pitch experiments were performed at these angles: $[-10^\circ, 0^\circ, 10^\circ, 20^\circ, 30^\circ]$. The signs of the angles are chosen according to Figure 10.

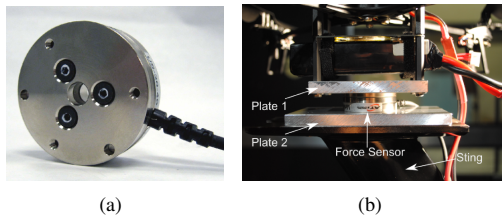


Figure 9: (a) Force sensor ATI Mini40-E and (b) a detailed view of the fixation of the quadrotor on the force sensor

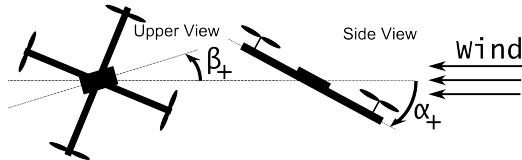


Figure 10: Positive Yaw and Pitch angles

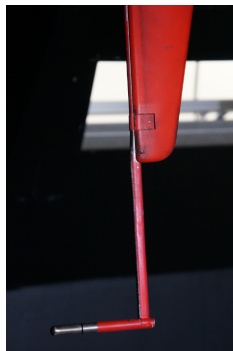
All the experiments were performed at three different wind speeds:

$$\text{Wind Speeds} = [4.7, 9.6, 12.9] \text{ m/s}$$

The wind tunnel has an integrated wind sensor (Figure 11.a) that is based on the pressure measurement on a sensitive surface, but this sensor is not fully reliable at such low speeds⁷.

For this reason, to have a double check of this measurement, a hot wire anemometer TMA-21HW was used (Figure 11.b). The main equations, on which the tests are based, are the following:

$$\begin{cases} F_{mx} = -D \cos \alpha \cos \beta - (L - mg) \sin \alpha \cos \beta \\ F_{my} = -D \cos \alpha \sin \beta - (L - mg) \sin \alpha \sin \beta \\ F_{mz} = -D \sin \alpha + (L - mg) \cos \alpha + |\mathbf{T}| \end{cases} \quad (24)$$



(a)



(b)

Figure 11: Wind Sensors: (a) Wind tunnel pressure sensor; (b) Hot wire anemometer TMA-21HW

⁷This wind tunnel is mostly used at its maximum wind speeds, which are around 50 m/s.

Where:

F_{mx} : measured force on the x axis of the force sensor [N]

F_{my} : measured force on the y axis of the force sensor [N]

F_{mz} : measured force on the z axis of the force sensor [N]

$\mathbf{T}$: Thrust vector [N]

D : Magnitude of Drag Force [N]

L : Magnitude of Lift Force [N]

α and β are two angles which give the orientation of the wind with respect to the body axes of the quadrotor. In the wind tunnel, the quadrotor is fixed and the wind is always blowing in the same horizontal direction. For this reason, the angles α and β of Equation 24 in the wind tunnel coincide with the pitch and yaw angles of the quadrotor.

It is assumed that the *Drag Force* has the following expression (according to the Rayleigh equation):

$$\mathbf{D} = \frac{1}{2} c_D A \rho V_w \mathbf{V}_w \quad (25)$$

where:

V_w : Wind Speed [m/s]

$\mathbf{V}_w$ is a unit vector giving the wind direction.

From the previous equation, it is clear that the *drag force* is a vector with magnitude $|\mathbf{D}|$ and with a direction that is always in the same direction of the wind vector $\mathbf{V}_w$.

Similarly to the Drag Force, it was assumed that the *Lift Force* has the following expression (according to the Rayleigh equation):

$$\mathbf{L} = \frac{1}{2} c_L A \rho V_w \mathbf{V}_{w\perp} \quad (26)$$

where $\mathbf{V}_{w\perp}$ is a unit vector perpendicular to the wind direction. From Equation 24, it follows that the identification of the $\mathbf{L}$ will be done through the observation of experiments at different pitch angles because it is related to the $\sin \alpha$.

The force sensor was not compensated for the force of gravity and for this reason that force is present in the previous equations.

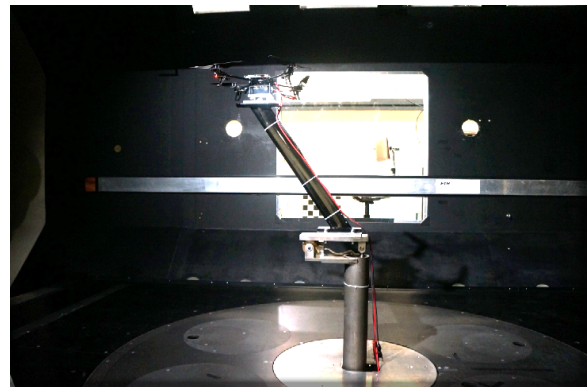


Figure 12: Quadrotor fixed in the wind tunnel

5.1 Yaw Experiments

While planning experiments in the wind tunnel, the approximation of the quadrotor to a cylinder sounded reasonable but, the tests proved that this is not true. From Figure 13, it is possible to see that the forces on the $x - y$ axis change according to the yaw angle variations and, for this reason, it is certain that there is a function that relates the c_{DA} to the yaw angle.

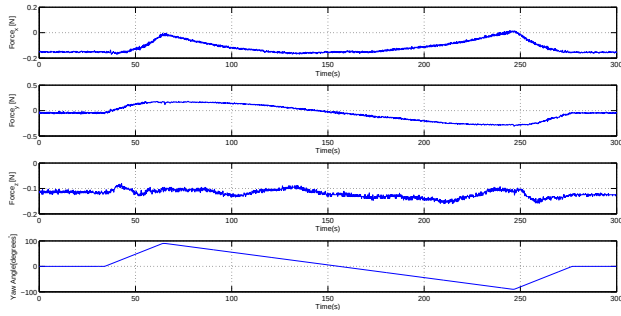


Figure 13: Forces in xyz with a $V_w = 4.7 \text{ m/s}$ and yaw variation

To analyze the data with $\alpha = 0$ and $\mathbf{T} = 0$, the system of Equations 24 becomes:

$$\begin{cases} F_{mx} = -D \cos \beta \\ F_{my} = -D \sin \beta \\ F_{mz} = (L - mg) \end{cases} \quad (27)$$

To have the drag force, only the 1st and the 2nd equation can be used. Solving the previous system of equations with respect to D will give:

$$D = \sqrt{F_{mx}^2 + F_{my}^2} \quad (28)$$

If the previous equation is applied to data corresponding to yaw angles $\beta \in [-90^\circ, 90^\circ]$, the result will be the $D(\beta)_{\beta \in [-90^\circ, 90^\circ]}$ drag force as a function of β . From the drag force, it is easy to extract the c_{DA} coefficient that will be, of course, a function of β :

$$c_{DA} = \frac{2D}{\rho V_w^2} \quad (29)$$

applying this inversion to the previous data of the drag force, the drag coefficient function $c_{DA}(\beta)$ showed in Figure 14⁸ for different wind speeds will be deduced.

Figure 14 draws attention to the poor reliability of the lowest wind speed experiment (4.7 m/s). This is due to the fact that the wind tunnel is usually used at high wind speeds because at such low wind speeds it can not ensure a laminar flow of the

⁸The density of the air ρ was accurately measured in the wind tunnel and on the day of the experiments was of 1.1283 Kg/m^3

wind. Indeed it is clear from Figure 14 that measurements at 9.6 and 12.9 m/s are the most reliable. They are quite similar to each other and they do not present the problem of different $c_{DA}(\beta)$ for $\beta = \pm 90^\circ$.

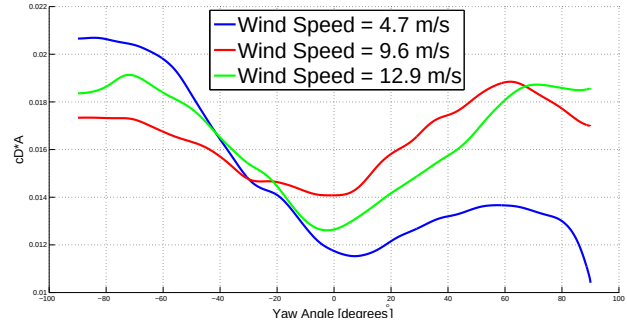


Figure 14: $c_{DA}(\beta)$ at different wind speeds

5.2 Pitch Experiments

In the pitch experiments, the yaw angle (β) is always equal to 0° . For this reason, the system of Equations 24 will become:

$$\begin{cases} F_{mx} = -D \cos \alpha - (L - mg) \sin \alpha \\ F_{my} = 0 \\ F_{mz} = -D \sin \alpha + (L - mg) \cos \alpha + |\mathbf{T}| \end{cases} \quad (30)$$

The pitch angles chosen for the experiments were the following: $[-10^\circ, 0^\circ, 10^\circ, 20^\circ, 30^\circ]$. Analyzing the data of the pitch experiments, the Tables 1-2 were obtained.

From the results in Tables 1-2, it can be assumed that the c_{DA} is a function only of the yaw angle (β) because it seems that this coefficient is not changing a lot as in the yaw experiments. This is a strong assumption and it should be validated with other experiments.

Another result from Tables 1-2 is that the c_{LA} is really small compared to the drag effect.

5.3 Thrust Experiments

Another category of experiments performed during this work were the thrust experiments, mainly for two reasons:

- 1 Identification of the *thrust function*: the function which relates the input δ given to the motors to the total thrust $|\mathbf{T}|$ they generate was refined.
- 2 Identification of the wind influence on the thrust vector: it was clear that the wind has an influence both on the direction and on the magnitude of the thrust vector $\mathbf{T}$.

Table 1: c_{DA} , c_{LA} at wind speed of 9.6 m/s and different pitch angles ($\beta = 0^\circ$ and $T = 0$).

Pitch Angle (α)	$V_w = 9.6 \text{ m/s}$	
-10°	$c_{DA} = 0.0143$	$c_{LA} = 0.0054$
0°	$c_{DA} = 0.0142$	$c_{LA} = 0.0017$
10°	$c_{DA} = 0.0119$	$c_{LA} = -0.0028$
20°	$c_{DA} = 0.0139$	$c_{LA} = 0.0018$
30°	$c_{DA} = 0.0119$	$c_{LA} = 0.0033$

Table 2: c_{DA} , c_{LA} at wind speed of 12.9 m/s and different pitch angles ($\beta = 0^\circ$ and $T = 0$).

Pitch Angle (α)	$V_w = 12.9 \text{ m/s}$	
-10°	$c_{DA} = 0.0144$	$c_{LA} = 0.0052$
0°	$c_{DA} = 0.0118$	$c_{LA} = 0.0048$
10°	$c_{DA} = 0.0146$	$c_{LA} = 0.0033$
20°	$c_{DA} = 0.0140$	$c_{LA} = 0.0002$
30°	$c_{DA} = 0.0141$	$c_{LA} = 0.0008$

As previously mentioned, the main result of the experiments performed in the wind tunnel was the identification of some terms of the aerodynamic model of the quadrotor such as c_{DA} and c_{LA} . In this way, it was possible to simulate the behavior of the quadrotor in case of wind in a more realistic way. After the identification of some terms of the aerodynamic model of the quadrotor such as c_{DA} and c_{LA} , it was possible to add wind into the simulator in a reasonable way. To do that, one choice is to add wind as a velocity vector with three components.

6 CONCLUSIONS AND FUTURE WORKS

In this work an approach to estimate wind and its influence on the state of a quadrotor was showed. During the work, an LQR/LQRI controller was implemented in order to control the position of the quadrotor, which allows it to reject constant disturbances.

With respect to the wind estimation, the main results are related to data coming from the wind tunnel experiments. They allowed us to perform a preliminary evaluation of the drag and lift coefficients adjusted by the area (A) of the section of the quadrotor interested by the wind flow. It was also possible to evaluate how wind speed and the pitch angle influence the thrust vector. Such aspects need further refinement and validation due to the few experiments that we were able to perform for the pitch variations.

For real flights the identification of the c_{LA} coefficient with turning propellers would be a future work of great interest because in that case they could act as wings and could have a large influence regarding the lateral in-flow.

This work could be a starting point for other projects addressing wind estimation on quadrotors, a crucial part of its overall control structure.

The main future works, beside those described above, which can be developed starting from this work are reported below:

- Validation of the results through real flights.

- Experiments in the wind tunnel with only one motor and one propeller.
- Identification of the full aerodynamic model of the quadrotor, and also analyzing the torque equations.

REFERENCES

- [1] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. Construction of cubic structures with quadrotor teams. In Hugh F. Durrant-Whyte, Nicholas Roy, and Pieter Abbeel, editors, *Robotics: Science and Systems*, 2011.
- [2] J Alonso-Mora, M Schoch, A Breitenmoser, R Siegwart, and P Beardsley. Object and animation display with multiple aerial vehicles. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [3] Caitlin Powers, Daniel Mellinger, Alex Kushleyev, Bruce Kothmann, and Vijay Kumar. Influence of aerodynamics and proximity effects in quadrotor flight. In *Proceedings of the International Symposium on Experimental Robotics*, June 2012.
- [4] Rahul Goel, Sapan M. Shah, Nitin K. Gupta, and N. Ananthkrishnan. Modeling, simulation and flight testing of an autonomous quadrotor. In *ICEAE*, 2009.
- [5] Konrad Rudin. Lecture Slides of the *Unmanned Aircraft Design, Modeling and Control* course: Dynamic Modeling of MAVs, 2013.
- [6] Anthony RS Bramwell, David Balmford, and George Done. *Bramwell's helicopter dynamics*. Butterworth-Heinemann, 2001.
- [7] Erdinc Altug, James P Ostrowski, and Robert Mahony. Control of a quadrotor helicopter using visual feedback. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 72–77. IEEE, 2002.
- [8] <http://www.pixhawk.ethz.ch>. Pixhawk, 2012.
- [9] <http://ardrone2.parrot.com/>. Parrot AR drone, 2012.
- [10] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar. The grasp multiple micro UAV testbed. *IEEE Robotics and Automation Magazine*, 2010.
- [11] MATLAB, version 8.2 (R2013b), The MathWorks Inc., 2013, Natick, Massachusetts.
- [12] Institut für fluidodynamik IFD of ETH, <http://www.ifd.mavt.ethz.ch/>, 2013.

Obstacle avoidance by combining background subtraction, optical flow and proximity estimation

Jaysinh Sagar and Arnoud Visser

ABSTRACT

For Micro Aerial Vehicles (MAVs), robust obstacle avoidance during flight is a challenging problem because only light weight sensors as monocular cameras can be mounted as payload. With monocular cameras depth perception cannot be estimated from a single view, which means that information has to be estimated by combining images from multiple viewpoints. Here we present a method which focuses only on regions classified as foreground and follow features in the foreground to estimate threat of potential objects being an obstacle in the flight path by depth segmentation and scale estimation. We evaluate results from our approach in an outdoor environment with a small quad-rotor drone and analyze the effect of the different stages in the image processing pipeline. We are able to determine evident obstacles in front of the drone with high confidence. Robust obstacle-avoidance algorithms as presented in this article will allow Micro Aerial Vehicles to navigate semi-autonomously in outdoor scenarios.

1 INTRODUCTION

In the domain of robot technology, we see a lot of progress using sophisticated hardware and software coming together to create a functional machine to accomplish varying tasks replicating human activity. Today, MAVs (Micro Aerial Vehicles) are used for tasks such as surveys, surveillance, search and rescue, and military applications which would otherwise be difficult or infeasible for humans considering harsher and challenging conditions. However, this is a challenging problem as activities that are simple and latent knowledge for living beings, are non-trivial when replicating on machines. Technology in this domain still has a way to go to achieve near human intelligence, but today, research tackles several sub-tasks which can someday be integrate into a system.

Specific kinds of drones have been developed such as flappy-wing drones, fixed-wing drones, and rotor based drones. Flappy wing drones have an advantage of being able to hover and fly close to objects being small, agile, and less dangerous due to their low weight. Fixed-wing drones are UAVs that use a gliding mechanism and use either external propulsion mechanism or built in linear propellers. Unlike

flappy wind drones, fixed-wing drones are not very agile but have longer flight durations and are extensively used for surveying and mapping applications.

In this study, we look at the challenge of obstacle avoidance using a quad-rotor drone in realistic environments with stationary obstacles. For this, we use the A.R.-Drone from Parrot S.A. which has the advantage of stabilized flight [1].

Previous approaches attempt to imitate biological approaches for obstacle detection and avoidance using optical flow in peripheral vision or stereo disparity from both eyes. However, these approaches are not suitable for oncoming obstacles directly in front of the camera. Scale based texture template matching algorithms mentioned in [2] may be used to detect on-coming obstacles but are subject to camera resolution and object texture. Supervised and semi-supervised approaches may be used as well if obstacle properties are known but may be challenging to provide in unknown environments.

The proposed method consists of a modular algorithm in order to generate confidence results using background/foreground classification, optical flow and filtering on affine relationships and scale estimation. The result of the algorithm generates a confidence mask used to direct and control the drone avoiding immediate obstacles. While clearly visible obstacles are effectively detected, performance of our method is subject to lighting and stability conditions. We employ existing methods of background/foreground classification and optical flow in conjunction with clustering, filtering and flow estimation methods to further improve avoidance results and effectiveness.

In section 2, we discuss related work in this area of study looking at advantages and limitations of these approaches. Section 3 touches upon general methods used in our obstacle avoidance application. Section 4 discusses the proposed method of this study followed by experiments, results and discussion in section 5. We then conclude and discuss future work in section 6.

2 RELATED WORK

In recent years, obstacle avoidance in MAVs has been a keen area of interest. Several approaches exist using monocular vision for detecting obstacle threats and avoiding such threats based on optical flow as well as using feature descriptors for relative size changes.

Bills *et al* [3] use an approach designed for indoor environments where there is uniformity in structure properties. The paper makes use of edges and detects lines using Hough transform and uses this information to classify the scene

based on a trained classifier. While the paper uses MAVs with a single camera, the proposed method in the paper allows the MAV to navigate in indoor environments where such features can be used. Using these features, a confidence value is determined based on line intersections crossing a visible vanishing point. Based on this confidence value, the MAV adopts a control scheme to react accordingly to its environment. When faced with an unknown environment; when confidence value is small; the MAV enters an unknown state where it adopts defined exploration strategies.

Celik *et al* [4] use an approach that exploits both lines and corner features detected in a scene. Similar to the Hough transform method, the paper uses line information extracted from the scene to determine the scene structure however, instead of running the method on the entire scene, as the Hough transform does, their method focuses only on scene parts where data exists. The next proposition of the paper is using a human like range estimation method using the height of the MAV and vanishing point of the scene. Using this information, the paper states that it is possible to obtain the relative distance of an object in the scene following feature points for its SLAM formulation. The paper uses a variation of optical flow as well for the proposed Helix Bearing Algorithm to create a motion field to determine turning points in the environment. This method is however also subject to camera capabilities and detection of strong features in the scene to navigate successfully.

Lee *et al* [5] talk about the use of MOPS (Multi-Scale-Oriented-Patches) and SIFT feature descriptors to obtain three-dimensional information of the environment. MOPS, similar to the Harris corner detector, is a quick and accurate corner point matching method. Using MOPS and SIFT, the paper attempts to reconstruct objects in 3D space where MOPS is used to extract edge and corner information and SIFT used to detect the internal outline information of an object. By combining these, a distinction between the outline and the inline of objects can be determined using matching distance between frames. By listing cases between the relationships of SIFT and MOPS features in the scene, it is possible to know the nature of objects around when the UAV moves and construct a 3D map of the environment. While the paper highlights experiments where they illustrate results from the proposed approach, additional knowledge of flight test data would help understanding the performance better.

Monocular cues used in the papers above ([3, 4, 5]) utilize information such as lines, corners and scene regularities to detect proximity and location of obstacles. However, the methods do not perform well in natural environments where such cues are challenging to find. Another limitation of these methods is that monocular approaches may not find distance to collision directly however [6] shows us that there exists a relationship between optical flow to time to collision which may be exploited to determine immediate threat of obstacles. Previous work on ground robots [7, 8, 9] use motion cues to

estimate depth from motion and background subtraction to determine obstacles in a frontal scene however face the issue of computational cost as well as robustness of detection. De Croon *et al* [10] use a combination of appearance variation cues and optical flow for their approach to obstacle detection. This method however is highly influenced by blob-based texture variations in the environment and has limited performance in outdoor scenarios.

The approach of More and Scherer [2] also combines features in conjunction with template matching to estimate scale, resembles the method in this study. The platform is also similar so a direct comparison of the results can be made. The approach of [2] allows the drone to estimate depth of objects detected as potential obstacles and observes relative changes in size of the sub-image template around that feature point to detect if the potential obstacle appears to come closer to the MAV by a fixed metric. While results of the study are positive for frontal obstacles using SURF features, a limitation of this approach is that performance is subject to presence of sufficient textures in the scene. In cases where there is not sufficient texture available to track (smooth and regular surfaces) the matching of SURF features do not perform as well as in the case of detailed textures. Also, the experiments carried out in this paper focus purely on narrow, tree-like obstacles in an indoor setting.

3 METHOD

The method studied in this paper uses background/foreground classification to find regions of interest in the MAV flight environment. Inside the region of interest potential obstacles are identified by a combination of optical flow, depth classification and scale template matching. The theory behind those techniques is worked out in the following subsections.

3.1 Background/Foreground classification

Background subtraction is a method in computer vision and image processing where an image is classified into the background and the foreground. This is done to achieve a variety of applications which require segmentation of concerned objects from a scene and filtering out everything else. Background subtraction creates a model of the background and then segments out everything that does not belong to the background as foreground based on a spatial and temporal setting. It can be seen as similar to detecting the objects are both in the new image and the old image. Once foreground objects are retrieved in the foreground mask, it is easy to extract as well as localize the objects in the scene.

There are several approaches to achieve background subtraction. Yet, due to the egomotion of our aerial vehicle a highly adaptive method is needed. Our method is inspired by the Improved Adaptive Background Mixture Model proposed by KaewTraKulPong and Bowden [11]. The method proposed is in two parts where we first consider background modelling and then maximizing the expectation over the

background Gaussian mixture model. The background is modelled using an Adaptive Gaussian Mixture Model where each pixel in the scene is modelled by K Gaussian distributions [12].

The probability of a pixel x_N at time N belonging to one of the background components is given as:

$$p(x_N) = \sum_{j=1}^K w_j \mathcal{N}(x_N; \mu_j, \Sigma_j) \quad (1)$$

where w_j is the weight parameter of the k^{th} Gaussian component $\mathcal{N}$ with mean μ_j and covariance Σ_j . Usually, if foreground objects are clearly distinguishable they will also be represented with a Gaussian component, but because there will be no evidence in the training set those components should have a small weight w_j . If the components are sorted on descending weights, the background can be modelled with minimal collection of components:

$$B = \underset{b}{\operatorname{argmin}} \left(\sum_{j=1}^b w_j > T_b \right) \quad (2)$$

where T_b is a threshold which corresponds with a prior estimate of the minimal portion of background objects one expect. The Gaussian Mixture Model is made adaptive by updating each component based on ownership o_j (the closest component with the largest weight w_j , as introduced by [13]) and a constant α which determines the learning rate, i.e. how fast old data is forgotten. The classical updates rules for each Gaussian component, in accordance with [12], for the update between timestep $t = N$ and $t = N + 1$:

$$w_j^{N+1} = (1 - \alpha)w_j^N + \alpha o_j^{N+1} \quad (3)$$

$$\mu_j^{N+1} = (1 - \alpha)\mu_j^N + \alpha \delta_j^{N+1} \quad (4)$$

$$\Sigma_j^{N+1} = (1 - \alpha)\Sigma_j^N + \alpha \delta_j^{N+1T} \delta_j^{N+1} \quad (5)$$

where $\delta_j = x - \mu_j$ is the distance from the pixel x to center of the Gaussian component μ_j . Although the adaptability of this algorithm can be tuned by the learning rate α , convergence could be too slow for the dynamics expected from a flying platform. Bowden *et al* [11] improved the algorithm by a window sampling method for faster convergence. In the L -recent window version the weight w_j^{N+1} is estimated from the previous L time frames, which gives the update rule:

$$w_j^{N+1} = \frac{1}{L} \sum_{t=0}^{L-1} o_j^{N+1-t} \quad (6)$$

The mean μ_j^{N+1} and covariance Σ_j^{N+1} of each Gaussian component are updated equivalently for the last L time frames.

With this approach, we have selected a method with could identify potential obstacles for the drone. Once we have the potential obstacles classified as foreground objects, we can use the tracking method explained in the next section to keep following those regions of interest.

3.2 Tracking and Depth Classification

We track features based on the classical Lucas-Kanade method for Optical Flow [14], pyramidal implemented as suggested by [15]. This method is based on the assumption that the flow of a concerned pixel in one image to another image should be present in the local neighborhood of that pixel in the first image. To recognize a equivalent pixels, independent of orientation and scaling, Lucas-Kanade describes a pixel with certain features. An algorithm that can provide adequate features is the Shi-Tomasi algorithm [16]. They propose the tracking of corners to avoid the aperture problem. Shi and Tomasi define a good feature in the same manner as Harris [17] which relies on the matrix of the second order derivatives of the image intensities. By calculating autocorrelation of the second derivative over small windows around each point we get a good description of the window. Shi and Tomasi found that a good corner could be easily described as long as the smaller of the two eigenvalues was greater than a minimum threshold.

Through optical flow, it is also possible to determine motion vectors or feature points that may be used to segregate faster moving features to slower moving features. For this segregation, we use clustering to condense the number of features in one region of the frame. To limit computation load we also added a neighbor count filter, which rejects isolated features. Our approach is inspired by building reciprocal nearest-neighbor chains [18], nicely described for the efficient average-link algorithm in [19]. The essence of this algorithm is as follows; one starts with a random feature and search for the nearest-neighbor. As long as the neighbor is close enough (relative to threshold θ), the neighbor is added to the chain. When the neighbor feature is already member of another chain, the two clusters are merged if they fulfill a *reducibility property* (the clusters are more similar to each other than any other cluster) [20]. When no close neighbor feature can be found a new random feature point is chosen to create a new chain. Note that the original approach [19] traverses through all combinations by sorting the all points from a common origin, while in our case a distance threshold is set. By calculating the average length of the optical flow for each cluster, it is possible to estimate affine relationships between closer and further obstacles, which can be used to evade closer obstacles.

3.3 Proximity estimation

In order to determine proximity of obstacles, calculating scale expansion of obstacles in the scene while moving closer makes it possible to estimate threat of collision. For this, we use a template matching approach based on *normalized*

cross-correlation [21] where we generate the previous image in different scales and then compare to the current image and find the best matching scale. For this, we take the mean location of dense feature groups from the result of tracking step and create a sub-window around that point from the previous frame. We then draw a sample template from the current image with the same location and dimensions and run the template matching algorithm to determine the relative change in scale.

3.4 Integration

To be able to robustly avoid obstacles, several methods (described in the previous subsections) have to be combined to generate control commands for the flying platform based on the images of the monocular camera. An overview is given in Figure 1. The control framework consists of two parts; the hardware side and the computer side.

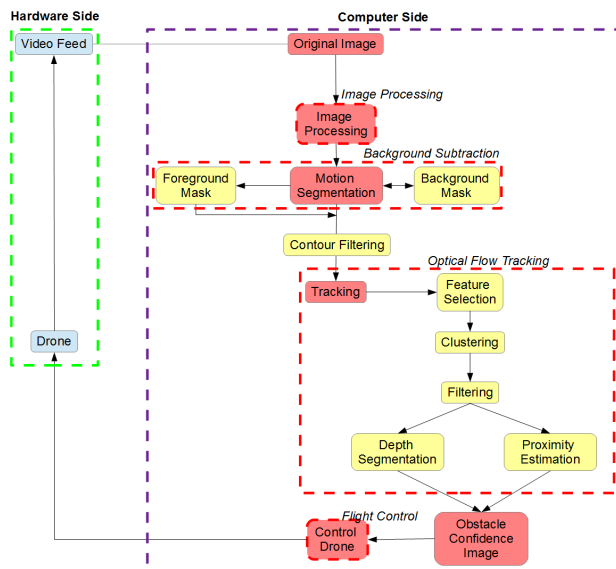


Figure 1: Overview of control pipeline

The final product of the image processing chain is an obstacle confidence image, which is used to steer the robot in a number of orthogonal directions (left, forward, right).

4 RESULTS

We conduct flight experiments in two situations to test performance of our method to highlight success and failure conditions. Flight path illustrations are plotted based on linear directions given by the drone controller to the drone in a sequence during the flight run. Example: [forward, forward, forward, left, left, left, left, forward, left, left, forward] where the drone moves by linear velocity at $\sim 1m/s$ in the given direction.

4.1 Two narrow objects as obstacles

In this experiment, the drone has to avoid two trees at a slight offset from one another where one is behind the other, as illustrated in Figure 2. In this scenario, an early decision to evade the second tree could result in a collision with the first tree. The experiment will be first performed with our complete algorithm, followed by an experiment without foreground, depth and proximity estimations.



Figure 2: Two trees as an obstacle in the flight path

As can be seen from Figure 3 that out of five runs, two runs avoid the trees with a flight path through the gap between the trees, two runs dodge both trees from the left side and one run dodges the first tree from the right side and does not encounter the second tree at all in its path. In four runs the flight path is not straight due to feature clusters detected from shadows on the ground. We also note in two runs (red and blue) there was a risk of the drone colliding sideways into first tree when avoiding the second tree.

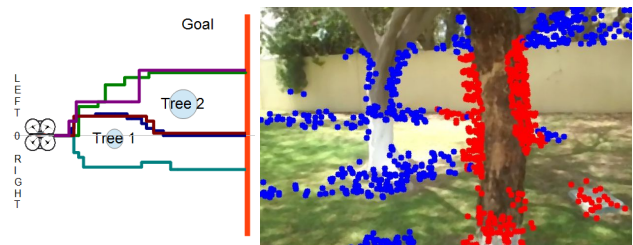


Figure 3: The path of the AR.Drone between two obstacles (left), based on the image including depth classification (right)

In Figure 4, we run the same experiment, yet this without filtering the optical flow in the background. Compared to the previous experiment, we see a large number of feature points detected on the trees in the background and not many on the obstacles. Flight results (Figure 4 (right)) do not have any successful results as features were detected everywhere in the scene except for the obstacles. As Tree 1 has more texture than Tree 2, two runs manage to successfully avoid Tree 1 however, the same level of accuracy is not achieved

with Tree 2 and feature points detected on the ground create a bias for the drone controller to fly left when it should be flying right resulting in collision with Tree 2. Three runs out of five do not detect enough features on Tree 1 resulting in collision. While a number of features detected may have detected features on the obstacles, we accept features only above a set score so that we do not have features with low scores. To summarize the impact of filtering of the optical flow; with BackGround/ForeGround classification 631 of 840 features can be found on the trees, while without this classification 159 of 1987 features are found on the trees.

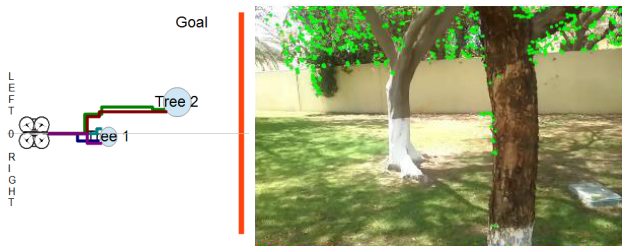


Figure 4: The path of the AR.Drone failing to avoid two obstacles (left) based on the image without ForeGround classification(right)

In Figure 5, we run an experiment using only proximity estimation. While the first tree is avoided well in all five runs, three runs fail due to collision with the second tree. This is due to the trunk of the second tree covers almost one half of the frame resulting in a poor template matching score. In two runs, the template matched with scale factor 1.1 of the previous frame texture and in one run, the frame was too obscured by the tree trunk due to insufficient score in the previous run. This may have been due to the fact that the second tree is wider than the first tree and there was not enough distance to measure the template accurately after avoiding the first tree.

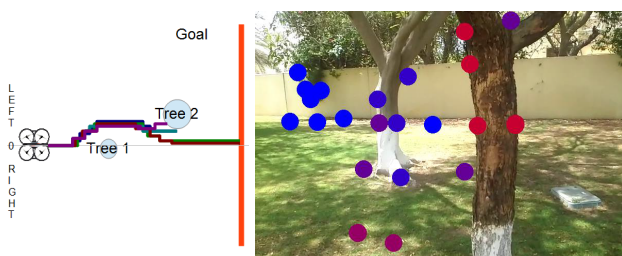


Figure 5: The path of the AR.Drone between two obstacles (left), without depth classification (right)

In Figure 6, we run an experiment using only optical flow without depth or proximity cues. All five flights in this setting are successful however the solution is a bit crude. As there is no depth perception, the process just detects both trees as an obstacle at the same degree and avoids them both immedi-

ately once detected. This method however is sensitive to feature patches found on the ground by tree shadows therefore we do not see a perfectly smooth flight once both obstacles are dodged.

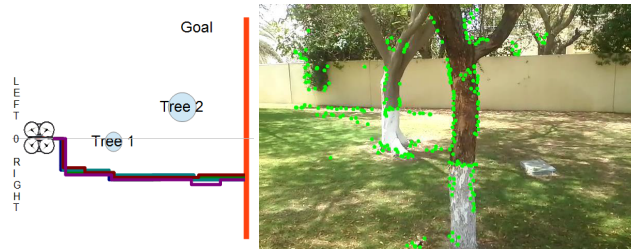


Figure 6: The path of the AR.Drone around two obstacles (left), with no depth nor proximity estimation (right)

4.2 Two wide objects as obstacles

In this experiment, we park two cars as wide body obstacles as a series of obstacles for the drone to fly around, as illustrated in Figure 7. This is another challenge; cars have a complete different texture and are so wide that they could fill the complete field-of-view. On the positive side; shadows are less prominent and the obstacles are more equally illuminated.



Figure 7: Two cars as obstacles in the flight path

Again we start the experiment with the full algorithm. As can be seen from Figure 8 there are four successful runs and one run (turquoise) failed. The failed run collided into first car just after taking off. We observe this to happen due to the template matching failing as the drone take off point was too close to the first car. The matching score was low causing the method assumed that the obstacle is far away. For the next four runs, we move the drone starting point a few steps back. Even after this step, the template matching score was not high enough however, the optical flow depth segmentation detected faster shift in feature points on the first car compared to detected features elsewhere in the scene due to the motion parallax property of optical flow. With this experiment, we see that when objects are too close to the drone on take-off,

the template matching algorithm fails as the drone does not observe a high degree of change in scale to satisfy the scale threshold in our method. Two runs (green and purple) successfully avoid first car however there is some delay in going around the right way when the second car is detected. This is due to a greater number of feature points detected on the rear of the car (right half of frame) compared to the front of the car (left half of the car). Once the drone decides to move left, features detected on the left side shift by a larger degree than the right side of the frame resulting in the controller giving the drone direction to move right successfully avoiding the second car.

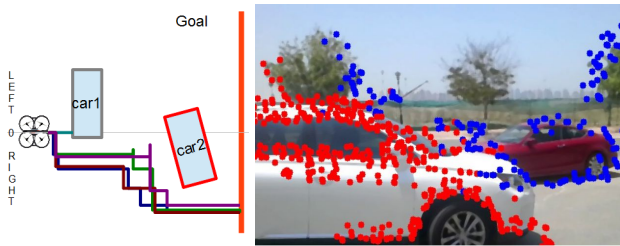


Figure 8: The path of the AR.Drone avoiding two cars (left), based on the image including depth classification (right)

In Figure 9, we see results of the proposed method without the foreground/background classification along with flight results. In the frames illustrated, we see that there are a lot of features detected on background trees and fence in the background of the scene. While there are a few features detected on the obstacles, most features are detected outside the two cars. Due to this, we see again a strong bias from background objects leading to non-perfect obstacle avoidance runs. Overall, we have two successful runs and three unsuccessful runs. When faced with car 1, two runs (red and green) detect a lot of features on the tree at the right side of the scene over features detected on the car 1. As not enough features are detected on car 1, the drone controller did not assume there to be an obstacle ahead resulting in collision. One run (purple) avoids car 1 and while crossing pass it, large number of incorrect features in the background and correct features detected on car 2 cause the drone controller to give instructions to move right colliding into car 1 along the drone's side. Two successful runs (blue and turquoise) avoid car 1 and get enough features on car 2 to successfully avoid it as well despite more features detected on other objects in the scene due to the depth and proximity estimations employed in the proposed method. If we analyze the number of correct features for this scenario, 874 of 1211 features are on the cars with background/foreground classification, while only 504 of the 1384 features are correct without this classification.

The experiment using the proximity estimation method without depth information confirms that this is no stable algorithm. As can be seen in Figure 10 all five runs were a failure.

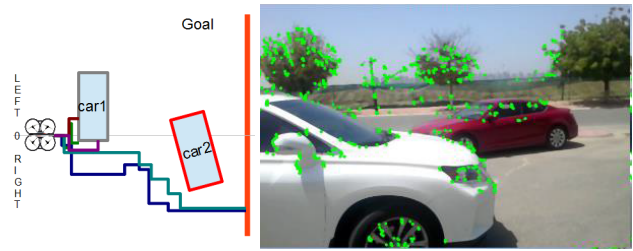


Figure 9: The path of the AR.Drone avoiding two cars (left), based on the processed image (right)

The first car was positioned very close to the starting position of the drone, so the initial template was not very descriptive. As a result the template matching routine always gave poor score values, so the flight controller just gave forward commands as no threats with high confidence were detected.



Figure 10: The path of the AR.Drone between two obstacles, without depth classification

In the experiment in Figure 11, we test the crude version of our algorithm. The runs are not very successful even though the method does detect feature points on the first car. For the three successful runs, the first car is avoided but going in the wrong direction. Even though the first car was close to the drone take off point, and feature points of the second car were detected as well, the scene contained greater number of feature points on the right side of the scene leading to the drone controller give the left command as all points appeared to be at the same level. Once the first car was crossed, the flight path lead to the right direction however avoiding the second car completely as it was no longer in the way. We also observe two paths (green and turquoise) not moving in any set direction at once point as the drone kept shifting left and right when in the center of the first car. This was because there were occasions where feature points on the first car kept going in and out of the scene as it was too close, the drone was stuck shifting left and right.

5 CONCLUSION

For each experiment carried out in this study, we see a good performance of the proposed method in a variety of situations. During the experiments, we see some significant influence of lighting and texture conditions where performance

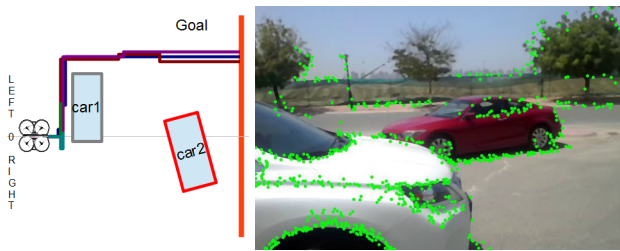


Figure 11: The path of the AR.Drone around two obstacles, with no depth nor proximity estimation

was noted to be better when there was good contrast between the obstacles and the surroundings. For outdoor experiments, we saw a lot of false positive features detected around trees in the background among others which caused a toll on the computation time required for optical flow between each frame pair. Due to which, we adjusted the contour threshold algorithm to have a higher contour threshold so as to eliminate small blobs caused by trees and leaves in the foreground mask from the background/foreground classification step. Feature detection is also significantly affected when there are not enough corner points found in large smooth parts of objects; e.g. car panels. Due to which the method has to rely entirely on corners and edges of the object. We also observe that accuracy of the template matching algorithm depends on regularity of illumination between frames. When there is a sharp change in illumination, the template matching algorithm returns low scores, and optical flow is also affected as feature are not matched. A failed run in the wide obstacle experiment shows us that, while the method performs adequately when the obstacle fits in the frame in its entirety, partially visible obstacles like the first car with low texture for a large part of the body cause an impact on the result. We also learn from comparison with other methods that while optical flow methods work, for obstacles and features moving from side to side, they do not function well when faced with walls or very narrow frontal obstacles. In the case with template matching, we see that it works fairly well when there is adequate distance between obstacles, but do not perform too well in tight navigation scenes. The proposed method having a combination of both methods performs better than them individually but is still subject to scene conditions of illumination and good features to track.

REFERENCES

- [1] Pierre-Jean Bristeau, François Callou, David Vissière, Nicolas Petit, et al. The navigation and control technology inside the ar.drone micro uav. In *18th IFAC World Congress*, pages 1477–1484, 2011.
- [2] Tomoyuki Mori and Sebastian Scherer. First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles. In *IEEE International Conference on Robotics and Automation (ICRA'13)*, pages 1750–1757, 2013.
- [3] Cooper Bills, Joyce Chen, and Ashutosh Saxena. Autonomous mav flight in indoor environments using single image perspective cues. In *IEEE international conference on Robotics and automation (ICRA'11)*, pages 5776–5783, 2011.
- [4] Koray Çelik, Soon-Jo Chung, Matthew Clausman, and Arun K Somani. Monocular vision slam for indoor aerial vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'09)*, pages 1566–1573, 2009.
- [5] Jeong-Oog Lee, Keun-Hwan Lee, Sang-Heon Park, Sung-Gyu Im, and Jungkeun Park. Obstacle avoidance for small uavs using monocular vision. *Aircraft Engineering and Aerospace Technology*, 83(6):397–406, 2011.
- [6] David N Lee et al. A theory of visual control of braking based on information about time-to-collision. *Perception*, 5(4):437–459, 1976.
- [7] Jeongdae Kim and Yongtae Do. Moving obstacle avoidance of a mobile robot using a single camera. *Procedia Engineering*, 41:911–916, 2012.
- [8] Abhijit Kundu, CV Jawahar, and K Madhava Krishna. Realtime moving object detection from a freely moving monocular camera. In *IEEE International Conference on Robotics and Biomimetics (ROBIO'10)*, pages 1635–1640, 2010.
- [9] Morimichi Nishigaki and Yiannis Aloimonos. Moving obstacle detection using cameras for driver assistance system. In *2010 IEEE Intelligent Vehicles Symposium (IV)*, pages 805–812, 2010.
- [10] GCHE De Croon, E De Weerd, C De Wagter, BDW Remes, and R Ruijsink. The appearance variation cue for obstacle avoidance. *IEEE Transactions on Robotics*, 28(2):529–534, 2012.
- [11] Pakorn KaewTraKulPong and Richard Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-Based Surveillance Systems*, pages 135–144. Springer, 2002.
- [12] Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.
- [13] Zoran Zivkovic and Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780, 2006.

- [14] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.
- [15] Jean-Yves Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker - description of the algorithm. Technical report, Intel Corporation, 2001.
- [16] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, 1994.
- [17] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.
- [18] J-P Benzecri. Construction d'une classification ascendante hiérarchique par la recherche en chaîne des voisins réciproques. *Cahiers de l'analyse des données*, 7(2):209–218, 1982. Rappel: *Cahiers de l'analyse des données*, 22(2):191-198, 1997.
- [19] Bastian Leibe, Aleš Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *International journal of computer vision*, 77(1-3):259–289, 2008.
- [20] Michel Bruynooghe. Méthodes nouvelles en classification automatique des données taxonomiques nombreuses. *Statistique et Analyse des données*, 3:24–42, 1977.
- [21] JP Lewis. Fast template matching. In *Vision interface*, volume 10, pages 120–123, 1995. The report "Fast normalized cross-correlation" expands the original paper.

Trajectory-aware Ad hoc Routing Protocol for Micro Aerial Vehicle Networks

Raheeb Muzaffar* and Evşen Yanmaz
Institute of Networked and Embedded Systems,
Alpen-Adria-Universität Klagenfurt, Austria

ABSTRACT

Micro Aerial Vehicles (MAVs) are small unmanned aerial vehicles (UAVs) that are generally equipped with camera, GPS and other sensors and are envisioned for many civil and commercial applications. Some of these applications require transmitting multimedia traffic and demand for a high wireless network throughput. In this paper, we consider an application scenario where a team of MAVs cover multiple areas of interest; e.g., during sports events, following known trajectories (mobility paths) and transmitting continuous streams of sensed traffic (images or video) to a ground station. We propose a Route Switching (RS) algorithm that utilizes both the location and the trajectory information of the MAVs to schedule and update routes to seamlessly transmit traffic to the destination. Simulation results show improved network performance in terms of throughput in comparison to Ad-hoc On demand Distance Vector (AODV) and Location Aided Routing (LAR) since the proposed algorithm exploits the added path information for route discovery.

1 INTRODUCTION

Numerous applications on aerial mapping, search and rescue, surveillance, transportation, etc. are identified [1, 2, 3, 4, 5, 6] where MAVs can be used with reduced cost and minimal infrastructure. Multiple MAVs can perform tasks faster and efficiently through coordination but would require strong wireless networking and communication capabilities in three-dimensional space [7]. In addition, Quality of Service (QoS) requirements are to be met to support these applications. The question arises if the existing widely accepted routing algorithms are sufficient to fulfill the requirements for a mission oriented network of MAVs [8].

In this paper, we consider a network of MAVs, where each MAV is equipped with Global Positioning System (GPS) and 802.11a wireless transceiver and is continuously streaming traffic to a ground station while following a pre-defined path in three-dimensional space. We propose a scheme that

schedules routing by exploiting the location and trajectory information of MAVs participating in the mission to improve the overall network performance. We investigate the network performance in terms of achieved throughput to evaluate the behavior of existing routing protocols namely AODV and LAR in a mission scenario and analyze if a simple route switching scheme can help in achieving better network performance and provide better support to fulfill QoS requirements of MAV applications.

The remainder of the paper is organized as follows. In Section II background on existing routing protocols namely AODV, LAR and Greedy Parameter Stateless Routing (GPSR) is summarized. Proposed route switching algorithm is presented in Section III. Section IV discusses the simulation setup and results and Section V concludes the paper.

2 RELATED WORK

A network of MAVs is associated with certain other constraints concerning routing compared to a ground wireless network such as high mobility, frequent topology changes, and routing in 3D space. AODV [9], LAR [10], Dynamic Source Routing (DSR) [11] and GPSR [12] are all reactive ad hoc routing protocols that require repetitive and exhaustive route discovery when a source tries to find a route to the destination. AODV broadcasts route request packets in the network to find the path to the destination. When an intermediate node receives a route request, it broadcasts the route request to its neighbors to find a valid route to the destination. Upon arrival of a route request at the destination node, the node sends a route reply using the reverse path. This way a route from source to destination is established.

LAR improves the flooding based route discovery mechanism by using the location information e.g. through GPS along with the flooding based scheme for route discovery. Source sends route request packet in the direction of the expected zone of the destination based on the location information. This directed broadcast reduces the number of packets required for route discovery and improves the overall network performance in terms of achieved throughput.

GPSR uses geographical location of the destination to route packets. Beacon signals are used for this purpose, which propagate location information of the nodes in the network. A source sends data to the destination by forwarding packets to its closest neighbor that is also closest to the destination, called the greedy forwarding mode. If the data packet

*Email address: raheeb.muzaffar@aau.at

reaches a region where greedy forwarding mode is not possible, the source node may shift to the perimeter forwarding mode to forward the data using planner graph traversal. A node using perimeter forwarding mode would shift back to the greedy forwarding mode whenever possible. Combining the greedy forwarding mode and the perimeter forwarding modes give the full GPSR algorithm.

In a multi-MAV environment the network topology changes frequently due to high mobility. The above mentioned protocols are not designed for such a setup [8] since the change in the network topology may lead to route disconnection from source to destination as the relay nodes no longer remain in the radio range. Subsequently, a route error message is sent to the source which then broadcasts a new route discovery message to find the new route to the destination. This consumes additional network bandwidth along with the delay in the traffic from the source to the destination. In this paper we propose a route switching scheme, explained in the next section that exploits the location and trajectory information to schedule routes from source to destination. In other words the route from the source to destination is calculated before the source loses its route to the destination due to a change in the network topology.

3 ROUTE SWITCHING

A trajectory-aware route switching mechanism is proposed to overcome the route error and the route discovery overhead of existing ad hoc routing protocols. The idea is to maintain information about the available routes from source to destination and switch to an alternate route when it is likely that the current route is going to break. We utilize prior knowledge on the position and mobility of nodes participating in the mission to switch to an alternate route.

The MAVs participating in the mission might need multiple hops to transmit their multimedia traffic to the destination. Since all MAVs are mobile, the established route being used to route packets to the destination may get disconnected. When this happens AODV or LAR will send a route error message to the source to which the source will initiate a new route request. However, using the prior knowledge of path information it can be predicted when a next hop MAV is going to go out of the communication range of an MAV that is part of the communication route from source to destination.

In such a case, the time for route error and thereafter a new route discovery can be avoided by switching to an alternate intermediate MAV before the current next-hop MAV goes out of the communication range. This is further illustrated in Fig. 1 where M_S is the source MAV, M_D is the destination ground node, M_{R1} and M_{R2} are potential MAVs that can relay from source to destination and M_N are other source neighboring MAVs. Here at time $T_{\alpha-1}$ route from source to destination is established via M_{R1} . We know that at time T_α , M_{R1} would no longer be in M_S radio range. Therefore, before time T_α the route from M_S to M_D can be switched via

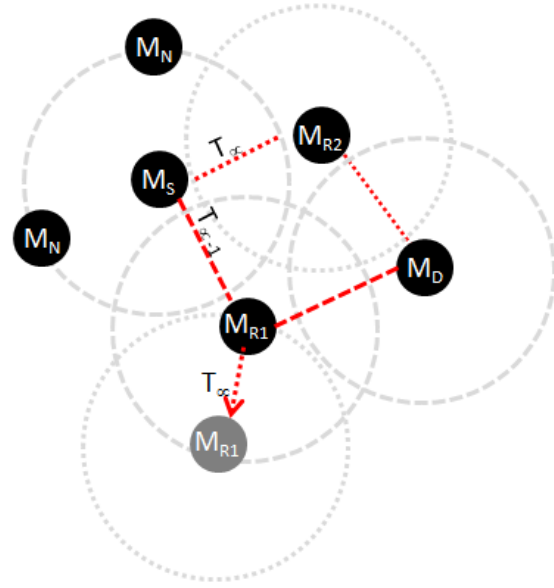


Figure 1: Transmission range and routes. The dashed and dotted circles and lines are the current and next transmission range and route respectively.

M_{R2} to avoid the time required for a new route discovery. In other words a new route can be established using the neighbor, location, trajectory and time information. This way the time for a new route discovery can be minimized as the communication link between M_S and M_D will remain established as long as an alternate neighbor M_{Ri} to relay is available.

3.1 Assumptions

1. Destination node is stationary on ground, all other nodes are moving in three-dimensional space.
2. The duration of the mission is defined and is known in advance to all nodes in the network.
3. In mission-oriented networks, where the team of MAVs operate together to achieve a goal, e.g., to continuously cover a known area, optimum pre-defined paths can be used [13]. Therefore, we assume that at any time instant of the mission, the source knows the location and trajectory of any node in the network through pre-defined path information.

3.2 Trajectory-aware routing protocol

The proposed routing-protocol is summarized in Algorithm 1. To find the route from the source to the destination, each source node first gains knowledge of its multi-hop neighbor nodes. The source then locates the destination node to find out if it can directly connect to the destination at any time instant during the mission. This is done using the trajectory information of the source and the destination. If so,

the source directly connects to the destination to transmit its data whenever possible and seeks the help of relay nodes otherwise. If a direct connectivity is not possible i.e., the destination is out of source communication range a multi-hop path has to be established. To do so, the source looks for its n^{th} hop neighbors that are connected to the destination. The n^{th} hop neighbors are sorted based on connectivity time with the destination. Now to select the n^{th} hop neighbor, the algorithm finds the $(n - 1)^{th}$ hop neighbors that are connected with n^{th} hop neighbors and so on until $(n - m)^{th}$ hop is the source itself. The connectivity time of all intermediate relay nodes is sorted. The calculated route is the one that provides the maximum connectivity time from the source to the destination. In other words, multiple routes are calculated but the route that has the maximum connectivity time is selected. However, since the source nodes are mobile, the calculated route can disconnect and so a new route must be calculated before the source experiences a disconnection. The algorithm thus calculates a new route as many times a disconnection is expected.

Algorithm 1 Trajectory-aware routing protocol

Input parameters: node trajectory and timing information, transmission range, channel model, source location.

1. Source acquires knowledge of its multi-hop neighbors
 2. Source checks if it will directly connect to the destination at any time instant during the mission
 3. Source routes its packets to destination directly whenever connected
 4. Otherwise, finds the n^{th} hop source neighbors that are connected to destination
 5. Sort connectivity time of the n^{th} hop neighbor with the destination and the $(n - 1)^{th}$ hop neighbor, select the most connected one
 6. Decrement n and repeat steps 4 - 6 until route is found
 7. Send packets to destination through the calculated route
-

There are some expected pros and cons involved. The proposed scheme benefits by providing an alternate route from the source to the destination during the mission; i.e., when a moving relay node gets out of the communication range of the source an alternate route without a new route discovery can be established to maintain connectivity. In other words, considering our mission, the expected delay in the multimedia transmission can be reduced through this scheme since the route error and route discovery overhead is avoided.

Nevertheless, it also comes with a cost of maintaining the knowledge of alternate routes. Routes are calculated based on the trajectories of the MAVs participating in the mission. The computation can be done centrally at the base station, which can then send the route information to each MAV or in a distributed way, where each MAV maintains trajectory

information of all the nodes in the network and calculates its own route accordingly. In either way, some storage capacity is required to maintain the trajectory information of the MAVs. The storage requirement can increase if the mission time is extended or more MAVs are added to participate in the mission.

Also, the computational cost to check when a relay node is going to get out of the range and when to shift to an alternate route is involved. More computational power is required to compute the routes as the network size increases or as the number of hops from the source to destination increase or both.

4 RESULTS AND DISCUSSION

Unless otherwise stated, the parameters used in the simulations are given in Table 1. These parameters are chosen in accordance with our platforms and experimental work [14]. We used Omnet++ as our simulation platform. Number of source MAVs chosen in the network are 1, 3, 6 and 9 respectively. Each source MAV continuously sends UDP traffic to the ground station at a rate of 54 Mbps such that the maximum channel capacity is utilized. The UDP packet size is set to 1480 bytes.

We investigate the performance of the proposed protocol for two mobility scenarios. First, we consider random mobility scenario where each node initially places itself randomly over the constrained area. Nodes then choose their destination randomly with random speed and direction. We then consider trajectories from a real coverage mission scenario of a disaster rescue operation [15]. The scenario considered in this paper is to provide live coverage through multimedia streams to multiple areas of interest where some sports events e.g., a marathon or a cycle race are taking place.

4.1 Random mobility scenario

First scenario uses random mobility model for hosts carrying camera, GPS, other sensors, etc. Random mobility model is popular due to its simplicity. To simulate the scenario we generated random trajectories shown in Fig. 2. We consider these generated trajectories as the area of interest for the MAVs to follow and transmit traffic to the ground station. Using this model a mobile host changes its direction uniformly randomly between $0^\circ - 360^\circ$, speed from 2 mps - 5 mps after a random interval of 5 s - 10 s. The destination node (FixedHost) denoted by ★ is kept stationary and all other nodes transmit while they are moving. Nodes move without any pause at any location. Three stationary relay nodes denoted by ♦ are also placed randomly to help route the traffic to the destination node. Stationary nodes are important since without them some source nodes e.g., MAV0, MAV6 and MAV8 are unable to form a route to the destination at many time instances during the mission time. Mobility of MAVs is constrained to an area of $1000 \text{ m} \times 1000 \text{ m} \times 50 \text{ m}$.

Table 1: Simulation parameters

Parameters	Values
Radio Interface	802.11a
Carrier Frequency	5 GHz
Number of Channels	1
Bit Rate	54 Mbps
Rate Adaptation	Adaptive Auto Rate Fallback
Mode	Ad-hoc
Channel Propagation	Free Space, Rayleigh
Transmission Power	7 dBm
Thermal Noise	-95 dBm
Radio Sensitivity	-90 dBm
Path Loss Alpha	2
Area bound	1000 m x 1000 m x 50 m
Simulation Time	900 s

We calculate achieved network throughput based on the number of data packets received during the mission time. The number of packets sent are more than the number of packets received since the source continuously transmits at a consistent rate of 54 Mbps while there may not be an available route to the destination. Packets not received due to broken or inaccessible link are dropped. The destination node is able to receive packets if the source is within its communication range or if there is a route available through relay nodes and has access to the communication channel. Figure 3 shows the achieved network throughput as the number of transmitting source nodes are increased from 1 – 9. We chose MAV0 (see Fig. 2) as the source node when 1 node is transmitting, MAV0, MAV1 and MAV2 when 3 nodes transmit and so on. Free space channel propagation model is used for this simulation. We observe that RS outperforms LAR and achieves approx. 10% higher network throughput since it utilizes the trajectory information to calculate the route from source to destination. This means that whenever a source gets a disconnection a new route (if available) is already calculated relinquishing the route error and route discovery overhead.

Figure 4 shows the corresponding packet inter-arrival time at the destination node i.e. the time interval between the received packets. We see less packet inter-arrival time for RS since as soon as the source desires to transmit, it already has the route (if available) to the destination and the transmission starts without a route discovery after initially acquiring knowledge of the neighbor nodes.

Figure 5 shows the achieved network throughput with increasing source nodes using Rayleigh channel propagation. Rayleigh fading model is reasonable when there is no line of sight between the sender and receiver and the incoming ra-

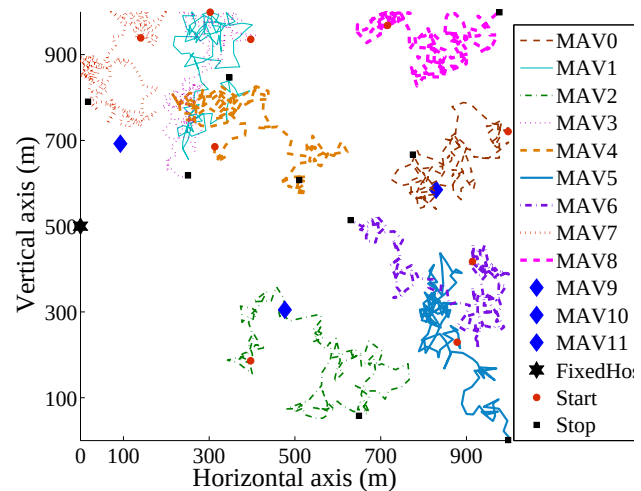


Figure 2: Trajectories: MAVs follow random paths

dio waves are received after being reflected or scattered by objects in the environment. We can observe that the overall achieved network throughput with Rayleigh fading is less compared to the free space model, which is expected but RS still outperforms LAR and achieves approx. 2% – 5% higher network throughput.

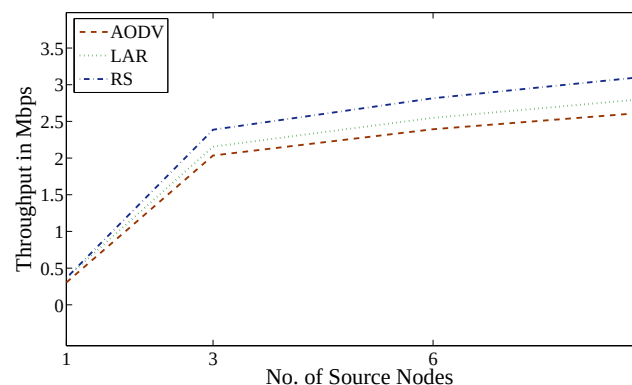


Figure 3: Achieved network throughput using free space channel propagation model

Until now, we have observed that the overall achieved network throughput for RS > LAR > AODV. We are now interested to evaluate individual performance of the MAVs in terms of the number of packets received and compare them using AODV, LAR and RS protocols. Figure 6 and Fig. 7 shows the cumulative sum of the received packets for MAV1 and MAV2 respectively, when three source nodes MAV0, MAV1 and MAV2 transmit simultaneously to the destination node. We chose MAV1 and MAV2 to evaluate the perfor-

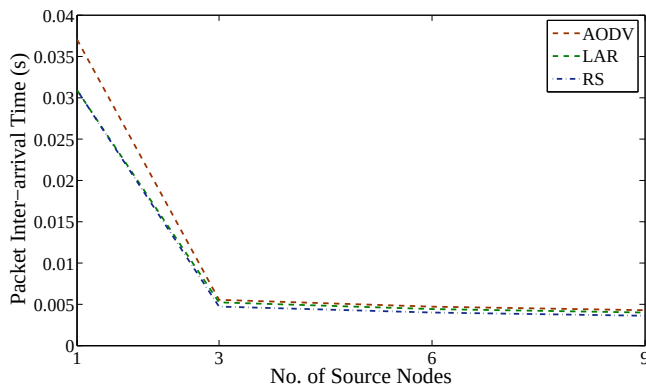


Figure 4: Packet inter-arrival time

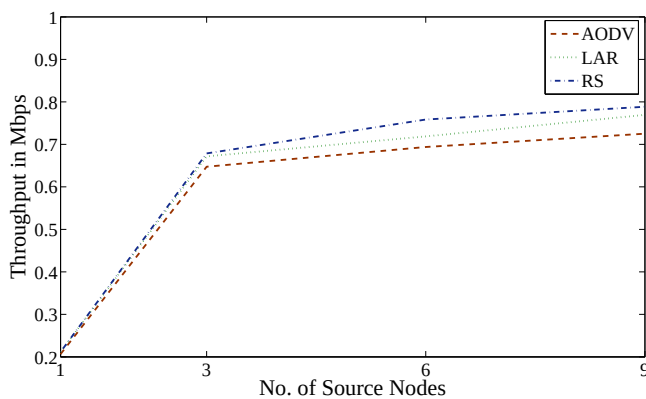


Figure 5: Achieved network throughput using Rayleigh channel propagation model

mance of a node (MAV1) that is relatively closer and requires less number of hops to connect to the destination and a node (MAV2) that it is relatively further and requires more number of hops for connectivity during the mission.

We observe that the individual comparison shows diversity at different time instances i.e., although RS performs better in terms of the sum of received packets and achieved throughput but at some time instances AODV or LAR performs better, e.g. the cumulative sum of received packets in Fig. 6 for MAV1 was better with AODV until 700 s and for MAV2 in Fig. 7 LAR performed better until 200 s. Although it depends on when a node gains access to the channel at a particular time instant but the results show that maintaining the route knowledge is helpful for achieving higher performance in terms of the total number of packets received. However, RS still lacks achieving better performance individually during the complete mission time. This means that there might be better possible links to the destination at some time instances that are not being utilized by the RS protocol. We so believe that better link throughputs can be achieved with a link aware routing algorithm. We intend to investigate this as our future

work.

Also, since our mission is to provide coverage to multiple events through multimedia traffic, we need to evaluate if the achieved throughput is sufficient to support such a scenario. From Fig. 3 the average network throughput is around 0.7 Mbps when three MAVs in the network transmit. In general the lowest quality MPEG video traffic requires 192 Kbps of data rate. Considering the throughput results achieved, it can be stated that with this setup multimedia traffic can be supported but the quality can be adapted at the application layer based on the available data rates at particular time instances. However, a good quality video link is required to be maintained during the mission time. Further improvement can be added with a link aware protocol and adding Quality of Service support at lower layers.

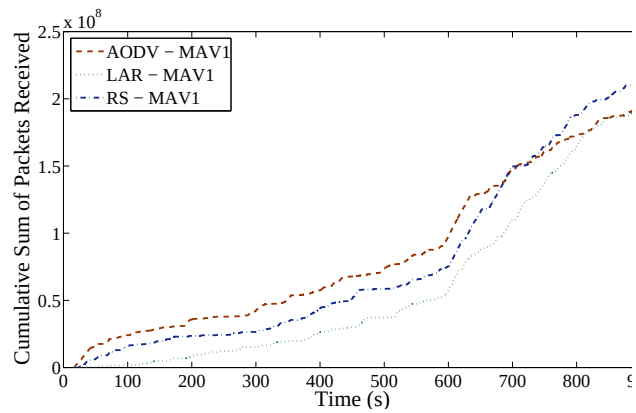


Figure 6: Cumulative sum of packets received from MAV1 while MAV0, MAV1 and MAV2 simultaneously transmit to the destination node

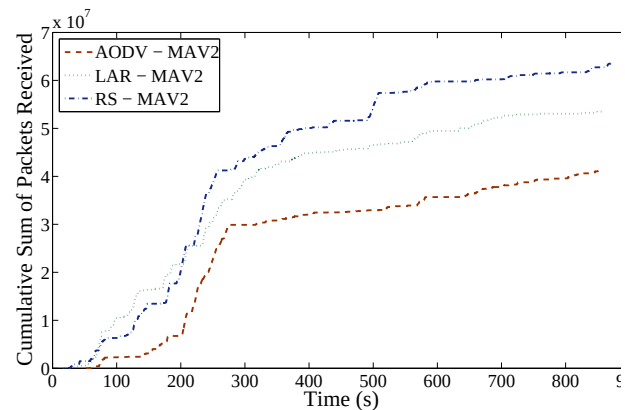


Figure 7: Cumulative sum of packets received from MAV2 while MAV0, MAV1 and MAV2 simultaneously transmit to the destination node

4.2 Coverage Mission

We now investigate the performance of our proposed protocol for the coverage mission scenario. The trajectories for this scenario that MAVs follow are computed while demonstrating a multi-MAV system that provides a high quality overview image of a given area of interest as shown in Fig. 8. The paths are then optimized based on the computed picture points [13] to provide maximum aerial coverage in the given mission time. The maximum mission time is set to 17 minutes considering the energy constraints of the quadrotors. The idea here to simulate using the real data set mission paths is that these paths resembles to our defined mission scenario of providing coverage through multimedia transmission to multiple areas of interest. The computed trajectories for the coverage mission scenario are shown in Fig. 9. Again, the destination node denoted by ★ is kept stationary and all other nodes transmit while they are moving. All other parameters are kept the same as given in Table 1 except for the simulation time which is set to 1000 s since some MAVs completes their path in this time.



Figure 8: Overview image of the area of interest for coverage mission

All MAVs start and stop at the same point and so in general they are initially and at the end closest to the destination node. Therefore, higher throughput is achieved at the start and at the end of the mission or whenever the node gets closer to the destination. It is also important to note that some paths are shorter than the others and so remain within the communication range of the destination. A MAV might thus only need two hops to transmit its packets to the destination and communicates directly otherwise.

Figure 10 shows the cumulative achieved network throughput using Free space propagation model for the coverage mission scenario. MAV0 (see Fig. 9) is the source node when 1 node is transmitting, MAV0, MAV1 and MAV2 when 3 nodes transmit and so on. The proposed Route Switching

algorithm still performs better than LAR although not much of a gain is visible. The reason here is that only two hops at max. are required for source to destination, otherwise the source connects directly to the destination. However, since the route is already calculated and known a slight improvement in the achieved throughput is noticed.

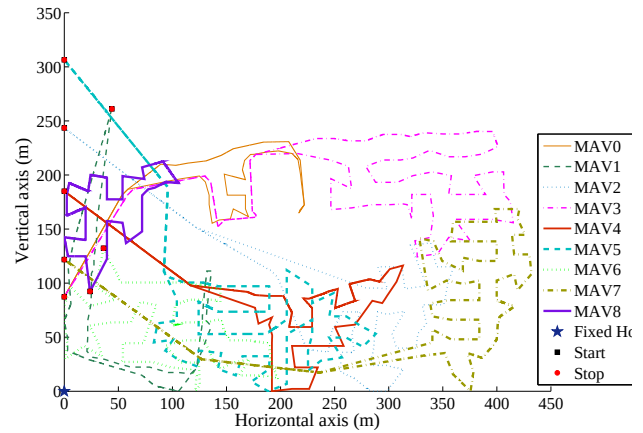


Figure 9: Trajectories: MAVs follow mission paths

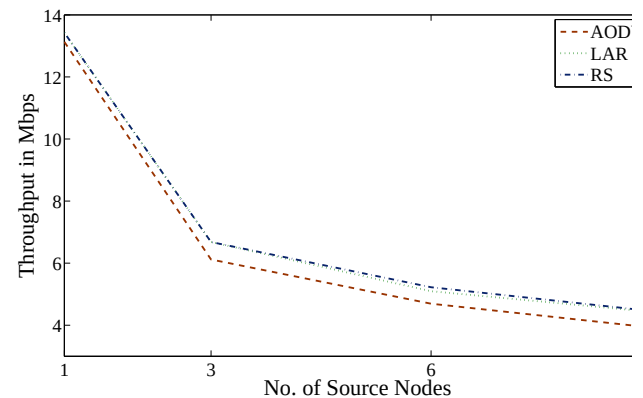


Figure 10: Achieved network throughput using free space channel propagation model for the mission scenario

5 CONCLUSION & FUTURE WORK

This paper analyses network performance in terms of achieved throughput and packet inter-arrival time of existing routing protocols for multi-MAV system to provide coverage to multiple areas of interest. We propose a route switching algorithm that exploits path information for calculating route from source to destination to overcome route discovery and route error overhead. However, this comes with the storage cost of maintaining the path information of all the nodes in the network and computational cost for calculating routes from each source to the destination. We used 802.11a radio

interface where each MAV continuously transmits traffic at 54 Mbps to the ground station. Simulation results show that the proposed route switching scheme outperforms LAR and AODV protocols by achieving higher network performance in terms of throughput where the trajectory information is known a priori. The performance can further be improved by adding QoS support to gain fairness among the nodes, link awareness at the network layer for better connectivity and adaptability at the application layer based on achievable data rates. Our future work will focus on evaluating the overhead cost of the proposed RS algorithm and developing methodology for a link aware routing in a multi-MAV networked environment.

ACKNOWLEDGMENT

This work was supported in part by the EACEA Agency of the European Commission under EMJD ICE FPA n° 2010-0012. The work has also been supported by the ERDF, KWF, and BABEG under grant KWF-20214/24272/36084 (SINUS). It has been performed in the research cluster Lakeside Labs.

REFERENCES

- [1] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley. Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use. *Remote Sensing*, 2012.
- [2] K. C. Wong. Survey of regional developments: Civil applications. In *UAV Australia Conf., Melbourne, Australia*, 2001.
- [3] M. Quaritsch, E. Stojanovski, C. Bettstetter, G. Friedrich, H. Hellwagner, B. Rinner, M. Hofbauer, and M. Shah. Collaborative microdrones: applications and research challenges. In *Proc. 2nd Intl. Conf. on Autonomic Comp. and Commun. Sys., Autonomics*, 2008.
- [4] A. Raptopoulos, P. Santana, D. Pachov, D. Damm, and I. Baruchin. [Online] We are creating the next paradigm for transportation using a network of unmanned aerial vehicles. <http://matternet.us/>, 2011.
- [5] H. McKenzie. [Online] Zookal starts world first delivery-by-drone service in Sydney. <http://pandodaily.com/2013/10/14/zookal-starts-world-first-delivery-by-drone-service-in-sydney/>, 2013.
- [6] S. Johnson. [Online] Flying donkeys. <http://www.flyingdonkey.org/>, 2013.
- [7] E. Yanmaz, R. Kuschnig, and C. Bettstetter. Achieving Air-Ground Communications in 802.11 Networks with Three-Dimensional Aerial Mobility. In *IEEE Proc. Intl. Conf. on Computer Commun. (INFOCOM), Mini conference*, 2013.
- [8] I. Bekmezci, O. K. Sahingoz, and S. Temel. Flying ad-hoc networks (FANETs): A survey. *Ad Hoc Networks*, 2013.
- [9] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Proc. Mobile Computing Sys. and Applications*, 1999.
- [10] Y. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. *Wireless Networks*, 2000.
- [11] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Proc. Mobile Computing*, 1996.
- [12] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. Intl. Conf. Mobile Comp. and Net., MobiCom*, 2000.
- [13] V. Mersheeva and G. Friedrich. Routing for Continuous Monitoring by Multiple Micro UAVs in Disaster Scenarios. In *Proc. European Conf. Artificial Intelligence*, 2012.
- [14] E. Yanmaz, S. Hayat, J. Scherer, and C. Bettstetter. Experimental performance analysis of two-hop aerial 802.11 networks. In *Proc. IEEE Wireless Commun. and Net. Conf. IEEE*, 2014.
- [15] Autonomous multi-uav systems. <http://uav.lakeside-labs.com/>, 2013.

Development of the flight model of a tilt-body MAV

L. R. Lustosa, F. Defay, and J. M. Moschetta *

ISAE - Institut Supérieur de l'Aéronautique et de l'Espace, University of Toulouse

ABSTRACT

This article presents the results of a wind tunnel campaign for a tilt-body UAV, the MAVion. The objective of this campaign is to develop a simplified flight model for use in control systems design and implementation. In order to achieve precise flight control during transition, stationary and cruise modes, the aerodynamic coefficients are identified for a wide flight envelope of angle of attack and sideslip. Additionally, the equilibrium transition is studied and the results validate the MAVion design. Moreover, an analysis of performance on aerodynamics due to addition of winglets in this platform is carried out.

1 INTRODUCTION

The use of Micro Air Vehicles (MAVs) for recognition missions in complex urban environments require the design of new MAV configurations capable of both horizontal and vertical flights. A typical mission scenario would consist of flying to a remote location, loiter and collect sensor data for an extended period of time, then hover to identify targets and possibly enter buildings, land, take off, without human assistance.

Typical fixed wing MAV configurations are appropriate to complete outdoor surveillance missions because they can cope with adverse windy conditions and provide a fairly good range as opposed to VTOL configurations. However, they are often limited in their practical application due to their intrinsic difficulty to sustain low-speed or hover flight. On the other hand, conventional VTOL MAV configurations such as multirotor platforms offer the capability to persistently survey an area by hovering or by achieving a perch-and-stare approach. However, VTOL configurations usually suffer from limited endurance and fail to efficiently sustain high speed flight. Therefore, it is of interest to combine the fixed and rotary wing capabilities into a single convertible-rotor aircraft.

In order to achieve either translation flight or vertical flight, different options are available. One is to directly tilt the rotors or the wing located in the rotor slipstream such as in the V-22 "Osprey" configuration. In the field of MAVs, the AVIGLE developed at Aachen University is an example of such a concept which requires an additional rotor above the horizontal tail in order to control the pitching moment [1]. Furthermore, it requires a tilting mechanism in the airframe

which leads to a significant weight penalty. Another approach consists of designing a fixed-wing configuration which can be tilted when needed so as to perform hover flight in a "prop hanging" mode. Such a tilt-body concept has been around for over half a century with the famous Convair XFY-1 "Pogo" developed and flight tested in the 1950s. Such a tail-sitter mini-UAV called "Vertigo" was developed and flown in 2006 at ISAE and further miniaturized in collaboration with the University of Arizona to provide the "Mini-Vertigo", a 30-cm span coaxial-rotor MAV capable of transition flight [2]. However, the coaxial rotor driving mechanism represents an additional weight and a technological limitation to design smaller versions. Furthermore, because of its limited aspect ratio, the Mini-Vertigo generates a fairly high induced drag in cruise conditions. In view of improving the aerodynamic performance in horizontal flight and simplifying the rotor mechanism, a new tilt-body configuration based on the bimotor flying wing has been designed. The MAVion has been initially designed to be a reasonably good airplane, capable of flying outdoors and easy to replicate as opposed to more complicated tail-sitters [3]. The main design guidelines were simplicity and transition flight capacity. The use of tandem rotors in tractor configuration allows for an additional degree of freedom which provides control in yaw and proved to ensure both hover and horizontal flight without resorting to complex tilting mechanisms [4]. The aerodynamic efficiency of the MAVion elevons is guaranteed over the whole transition flight range since in hover, when the freestream flow is drastically decreased, the propeller slipstream maintains the elevator aerodynamic efficiency. Previous studies have investigated the problem of a propeller at very high angle of attack [5] and the propeller-wing interaction [6, 7] while few studies have been conducted to actually provide a full model of the aircraft in transition flight. It is the purpose of the present paper to describe the development of a full aircraft model based on a new wind tunnel campaign in which a broad range of angles of attack and slip angles have been tested.

2 TILT-BODY MAV DYNAMICS MODELING

2.1 Aerodynamic forces and moments

Aerodynamic forces and moments are described in this work in 2 different orthogonal coordinate systems, namely, body-fixed (Figure 1) and wind systems. An inviscid incompressible flow is assumed and, consequently, the aerodynamic coefficients for the MAVion $C_{\square}(\alpha, \beta, \delta_1, \delta_2)$ are dependent on angle of attack α , sideslip β and elevons deflection δ_i , and

*Email address(es): (leandro-ribeiro.lustosa, francois.defay, jean-marc.moschetta)@isae.fr

are related to aerodynamic forces and moments by

$$\begin{aligned} F_x &= C_x q_\infty S & M_x &= C_{mx} q_\infty S b \\ F_y &= C_y q_\infty S & M_y &= C_{my} q_\infty S \bar{c} \\ F_z &= C_z q_\infty S & M_z &= C_{mz} q_\infty S b \end{aligned} \quad (1)$$

and

$$\begin{aligned} F_D &= C_D q_\infty S & M_d &= C_d q_\infty S b \\ F_Y &= C_Y q_\infty S & M_y &= C_y q_\infty S \bar{c} \\ F_L &= C_L q_\infty S & M_l &= C_l q_\infty S b \end{aligned} \quad (2)$$

where

$$q_\infty = \frac{1}{2} \rho_\infty V_\infty^2 \quad (3)$$

and q_∞ , ρ_∞ and V_∞ denote, respectively, free-stream dynamic pressure, density and velocity, whereas S denotes MAVion reference area defined by the product of its mean aerodynamic chord $\bar{c}$ and its wing-span s . Moments in both coordinate systems are taken with respect to the center of mass.

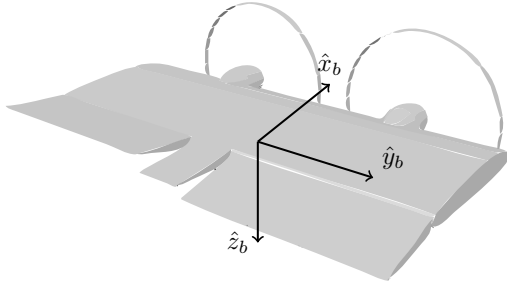


Figure 1: Perspective view and body-axis definition.

In the present work, the $C_x, C_y, C_z, C_{mx}, C_{my}, C_{mz}$ aerodynamic coefficients are identified by means of a wide-envelope wind tunnel campaign in α and β and can be related to wind-fixed coordinates by a rotation transformation, namely

$$D_w^b = \begin{bmatrix} -\cos(\beta)\cos(\alpha) & -\sin(\beta) & -\cos(\beta)\sin(\alpha) \\ -\sin(\beta)\cos(\alpha) & \cos(\beta) & -\sin(\beta)\sin(\alpha) \\ -\sin(\alpha) & 0 & -\cos(\alpha) \end{bmatrix} \quad (4)$$

so that

$$\mathbf{C}_w \triangleq \begin{pmatrix} C_D \\ C_Y \\ C_L \end{pmatrix} = D_w^b \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} \triangleq D_w^b \mathbf{C}_b \quad (5)$$

and

$$\mathbf{C}_{m,w} \triangleq \begin{pmatrix} C_d \\ C_y \\ C_l \end{pmatrix} = D_w^b \begin{pmatrix} C_{mx} \\ C_{my} \\ C_{mz} \end{pmatrix} \triangleq D_w^b \mathbf{C}_{m,b} \quad (6)$$

2.2 Propulsion forces and moments

Similarly to the aerodynamic model development, a dimensional analysis is performed assuming little influence of viscous effects and Froude number and it leads to MAVion forces T_i and moments Q_i applied at the propeller i given by

$$T_i = \rho n_i^2 D^4 C_T(J_i) \quad (7)$$

and

$$Q_i = \rho n_i^2 D^5 C_Q(J_i) \quad (8)$$

where

$$J_i = \frac{V_{\infty,N}}{n_i D} \quad (9)$$

and n_i , D , C_T , J_i denote motor speed in revolutions per second, propeller disk diameter, coefficients of thrust and moment, and advance coefficient. C_T and C_Q are modelled as a linear relation in J_i as previously studied in [8].

2.3 Propeller downwash effects

In horizontal flight, the propeller system pushes air downstream in order to balance small drag forces and creates a downwash slipstream towards the aerodynamic surfaces which can usually be disregarded due to its small strength and influence on the overall forces system. However, in transition mode or vertical flight, a more influent downwash slipstream is created to overcome the weight of the aircraft. This slipstream interacts with the aerodynamic surfaces and creates induced forces and moments.

The fundamental idea applied in this work is to divide the aerodynamic surfaces in two parts, namely, wet and dry parts (see Figure 2). The dry part has an area S_{dry} whereas the wet part area S_{wet} is such that, by the disk actuator theory,

$$S_{wet} = \frac{1}{2} S \quad (10)$$

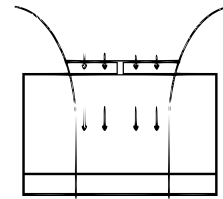


Figure 2: Propeller wake illustration.

Independence between areas S_{dry} and S_{wet} is assumed such that the dry area is unaffected by the propulsion system and its empirically modeled in the light of the blade element theory as

$$\mathbf{F}_b = q_\infty S_{dry} \mathbf{C}_b(\alpha, \beta, \delta_1, \delta_2) \quad (11)$$

and

$$\mathbf{M}_b = q_\infty S_{dry} \begin{bmatrix} b & 0 & 0 \\ 0 & \bar{c} & 0 \\ 0 & 0 & b \end{bmatrix} \mathbf{C}_{m,b}(\alpha, \beta, \delta_1, \delta_2) \quad (12)$$

whereas the wet part is computed as

$$\mathbf{F}_b = q_{\infty, ind} S_{dry} \mathbf{C}_b(\alpha_{ind}, \beta, \delta_1, \delta_2) \quad (13)$$

and

$$\mathbf{M}_b = q_{\infty, ind} S_{dry} \begin{bmatrix} b & 0 & 0 \\ 0 & \bar{c} & 0 \\ 0 & 0 & b \end{bmatrix} \mathbf{C}_{m,b}(\alpha_{ind}, \beta, \delta_1, \delta_2) \quad (14)$$

where α_{ind} and $q_{\infty, ind}$ capture the influence of the wake flow on the aerodynamic surfaces and are modelled as [2]

$$\alpha_{ind} = \alpha - \sin^{-1} \left(\frac{w \cos \beta \sin \alpha}{V_{\infty, ind}} \right) \quad (15)$$

and

$$q_{\infty, ind} = \frac{1}{2} \rho V_{\infty, ind}^2 \quad (16)$$

where $V_{\infty, ind}$ is given by

$$V_{\infty, ind}^2 = (w + V_{\infty} \cos \beta \cos \alpha)^2 + (V_{\infty} \sin \beta)^2 + (V_{\infty} \cos \beta \sin \alpha)^2 \quad (17)$$

and w is the solution of the following quartic

$$w^4 + 2V_{\infty} \cos \alpha \cos \beta w^3 + V_{\infty}^2 w^2 = \left(\frac{2T_i}{\pi \rho D^2} \right)^2 \quad (18)$$

3 WIND TUNNEL CAMPAIGN

Due to the convertible nature of the vehicle, a 6-component (3-dimensional forces and moments) wide angle of attack and sideslip envelope wind tunnel campaign took place to validate the 6 degree-of-freedom theoretical model.

3.1 SabRe Wind Tunnel and Balance

The experiments were ran at the SabRe closed-loop wind tunnel (Figure 3) located at ISAE and capable of delivering low Reynolds stable and uniform flow at a wind velocity range of 2 to 25 m/s, thus ideal for experimenting full-span micro air vehicles.

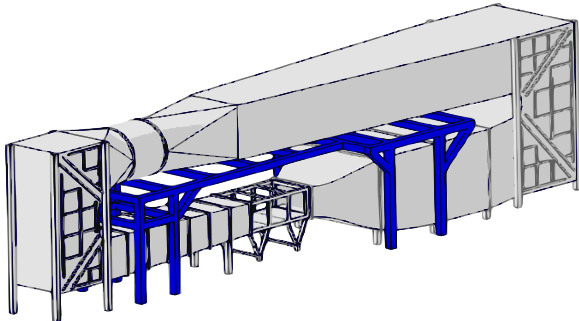


Figure 3: SabRe Wind Tunnel.

Forces and moments were measured by means of a 5-component internal balance (2-dimensional forces and 3-dimensional moments measured in the balance body coordinate system). Therefore, the desire of a full 6-component study calls for 2 different internal balance configurations (see Figure 4) for every set of experimented $\{\alpha, \beta, V_{\infty}, \omega_1, \omega_2, \delta_1, \delta_2\}$. Therefore a duplicity of values for $\{F_{bx}, M_{bx}, M_{by}, M_{bz}\}$ arrives and it is used for evaluating repeatability and detecting procedural errors.

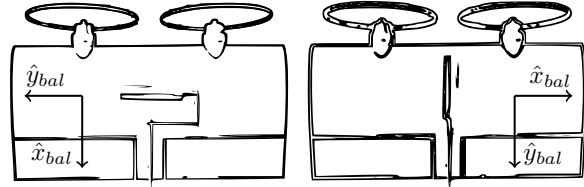


Figure 4: Two internal balance configurations.

Comparing the MAVion and balance coordinates systems (Figures 1 and 4) in both configurations, and assuming MAVion and balance perfect alignment, it can be readily seen that

$$\begin{pmatrix} F_{bx} \\ F_{by} \\ F_{bz} \end{pmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (19)$$

and

$$\begin{pmatrix} M_{bx} \\ M_{by} \\ M_{bz} \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} L \\ M \\ N \end{pmatrix} \quad (20)$$

for balance configuration 1, whereas

$$\begin{pmatrix} F_{bx} \\ F_{by} \\ F_{bz} \end{pmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (21)$$

and

$$\begin{pmatrix} M_{bx} \\ M_{by} \\ M_{bz} \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} L \\ M \\ N \end{pmatrix} \quad (22)$$

for balance configuration 2. The opposite signs in force and moment matrix transformation are due to balance manufacturer left-hand moment convention notwithstanding MAVion right-handed coordinate system.

The MAVion model is rigidly mounted on a movable platform with 2 degrees-of-freedom with respect to the wind, namely, yaw θ_1 and pitch θ_2 such that

$$\begin{cases} -\frac{\pi}{2} < \theta_1 < \frac{\pi}{2} \\ -\frac{\pi}{6} < \theta_2 < \frac{\pi}{6} \end{cases} \quad (23)$$

due to mechanical restrictions. Since the flight envelope includes horizontal flight (θ_2 near 0) and vertical flight (θ_2 near

$\pi/2$), an additional wind tunnel set-up difficulty arises due to limitations in the total excursion of θ_2 . Such shortcoming calls for additional testing configurations illustrated by Figure 5. In configuration M, large angles of sideslip are achievable and the following direct relations between wind and platform angles can be written

$$\alpha = \theta_2 \quad (24)$$

and

$$\beta = \theta_1 \quad (25)$$

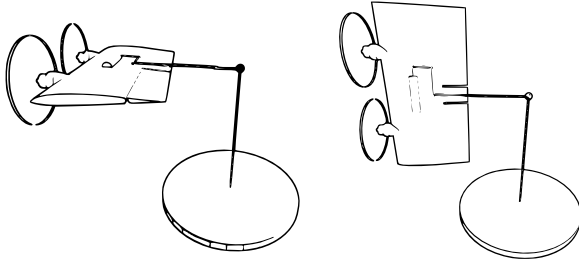


Figure 5: Wind tunnel configurations. On the left, configuration M. On the right, configuration M'.

Complementary, configuration M' delivers high angles of attack, but the relation between wind and platform angles is not obvious anymore, and a series of equivalent transform rotations will deliver

$$\alpha = -\tan^{-1}(\tan\theta_1 \sec\theta_2) \quad (26)$$

$$\beta = \sin^{-1}(\cos\theta_1 \sin\theta_2) \quad (27)$$

Finally, several configurations in this study suffer from potential ground and blockage effects due to high angles of attack. Previous work demonstrated that ground effects are negligible, and as so, they will be disregarded herein. However, future theoretical and computational studies are planned to account for forces and moments errors due to blockage in this campaign.

3.2 Wind Tunnel Model and Acquisition System

An adapted MAVion was manufactured for wind tunnel campaign purposes (Figure 12). Its main objectives were to enable rigid installment of the internal balance in both configurations and to provide a non-deformable airfoil section to aerodynamic identification. Figure 6 illustrates the electronics counterpart.

Two servo-motors are responsible for elevons deflection. Two potentiometers are installed in both elevons as to accurately measure elevon deflections δ_i . Notice that the potentiometer inside the servo dedicated to move the elevon could perform similar task, but it is more inaccurate due to rod deformations and servo-control errors.

The propulsion system, namely, two brushless CC motors equipped with speed controllers, which communicate by I2C

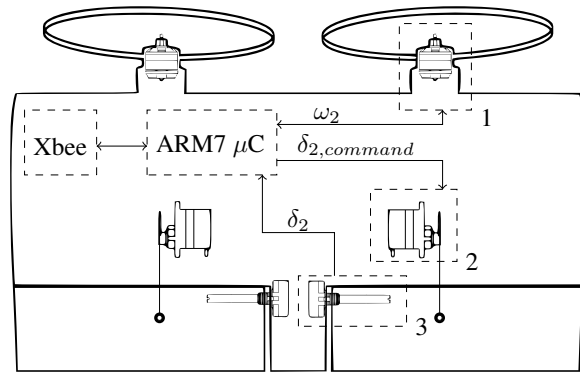


Figure 6: MAVion wind tunnel model instrumentation.

with MAVion onboard computer, controls and measures the rotation speed ω_i of each helix.

The wind tunnel ground station (Figure 7) communicates with MAVion onboard computer by means of a wireless XBee connection in order to minimize cables and their induced forces and torques on the system due to their intrinsic stiffness. The wind tunnel ground station is then responsible for commanding motor speeds and elevons deflections in an open-loop configuration, receiving telemetry data from MAVion (ω_i and δ_i , which may be different from commanded values due to the open-loop control nature), receiving data from balance and registering all measured values.

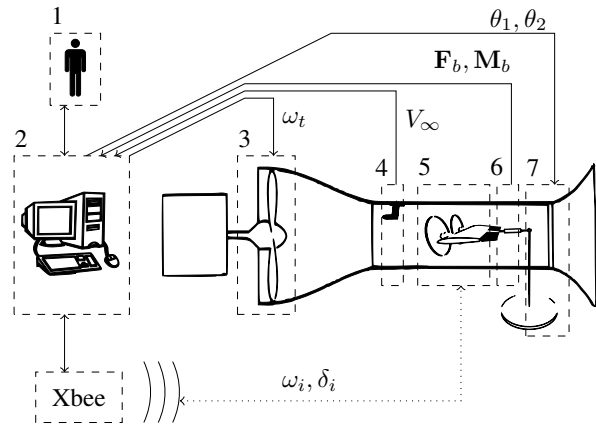


Figure 7: Wind tunnel acquisition system set-up.

Two distinct methodologies were used to collect wind tunnel data, namely, equilibrium transition study and exploration study. In both, the wind tunnel operator set the operating conditions of tunnel and MAVion, collected sensors measurements and registered them all in a line of a table corresponding to an experimental trial. The operating conditions included wind velocity V_∞ , angle of attack α and sideslip β , elevons deflection δ_i , propulsion motors speed ω_i . The sen-

sors measurements included flow temperature and pressure, precise flow speed, elevons potentiometers readings, propulsion motors controllers rotational speed, and 5-component balance forces and moments.

3.3 Equilibrium Transition Study

Equilibrium points, i.e. flight condition points $(\alpha, \beta, V_\infty, \delta_1, \delta_2, \omega_1, \omega_2)$ in which the sum of gravitational, aerodynamic and propulsion forces and moments are null, are particularly interesting for analyzing flight envelope and steady flight conditions. By means of the wind tunnel campaign, it will be experimentally shown that exists a n-tuple $(V_\infty, \delta_1, \delta_2, \omega_1, \omega_2)$ such that longitudinal equilibrium is achieved for each α throughout the entire $[0^\circ, 90^\circ]$ angle of attack envelope and thus the MAV design allows for a quasi-static¹ equilibrium transition between horizontal and vertical mode. Therefore, when there's no wind, transitioning between horizontal and vertical modes can be achieved by climbing over the target until the relative ground speed reaches zero (Fig. 8). However, when there is some wind, hovering with respect to a target on the ground requires an equilibrium flight with non-zero vehicle velocity with respect to the wind (Fig. 8) and the ability to hover in windy conditions is limited by the vehicle maximum velocity with respect to the wind.

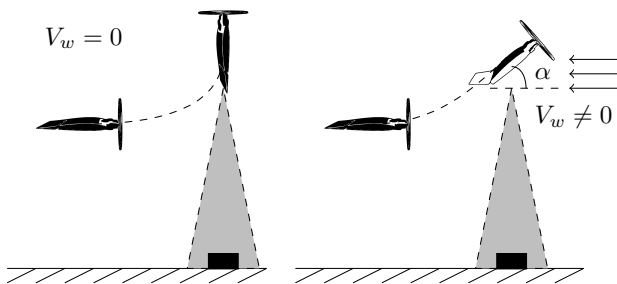


Figure 8: Quasi-static equilibrium transition maneuvers and hovering over ground target under non-windy (left) and windy (right) conditions.

Equilibrium points were studied in this campaign by following the general guidelines stated by [4] for longitudinal flight, i.e., $\beta = 0$. The fundamental idea is to, for each sampled α , search for values of $V_\infty, \delta_i, \omega_i$ that will deliver aerodynamic/propulsion forces and moments (notice gravitational suppression) that will cancel gravitational forces and moments on the flying model, which has a different mass distribution than the wind tunnel model that is not meant to fly. The desired flying model mass distribution is such that the

¹Quasi-static equilibrium transition additionally allows us to disregard aerodynamic derivatives effects in the MAVion dynamic model. However, if a faster transition is required, the aerodynamic derivatives effects can no longer be neglected and should be included in the differential equations that model the system. They also should be identified by theory, experimental flights or a different type of wind tunnel campaign.

center of mass is longitudinally located at a position $0.15c$ away from the leading edge and is motivated by a 10% stability margin controls requirement.

At each α , flow velocity was modified until lift was balanced by weight. Next, the propulsion motors velocity was set such that drag was equal to thrust. Then, the elevons were deflected in such a way to eliminate pitching moment. However, modifying one variable unbalances the other equilibria and an iterated scheme is proposed for reaching longitudinal equilibrium. Once it is found, all wind tunnel data is recorded as a new point in the equilibrium transition experimental table. Current and voltage measurements also took place to study power requirements. The results can be seen in Figure 9.

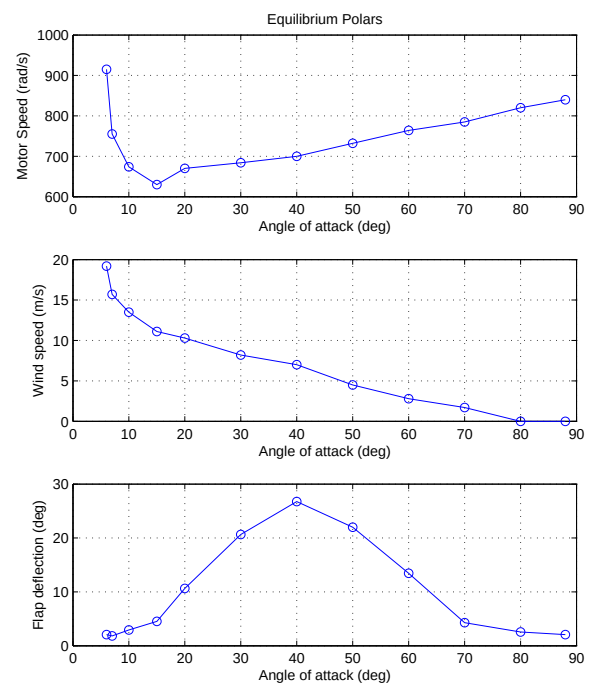


Figure 9: Equilibrium transition study.

The results show that the vehicle is actually capable of sustaining equilibrium flight during the whole transition phase between 0 and 20 m/s, which means that it is capable of hovering with respect to the ground for lateral wind speeds ranging from 0 to 20 m/s (Figure 9B). Figure 9A indicates that the motor regime is limited by two extreme cases: hover at 90° angle of attack and fast cruise speed at 5° angle of attack. Finally, the elevator deflection angle is plotted as a function of the angle of attack, showing a maximum deflection value of 27° at 40° angle of attack, within the range of the elevator aerodynamic efficiency. This confirms that because the elevons are placed within the propeller slipstream,

control in pitch is maintained throughout the entire transition flight phase.

3.4 Exploration Study

Data were also taken by independently setting the angle of attack, sideslip, motor speed and flap deflection to the values illustrated by tables 1 and 2. All combinations were exhaustively tried and the associated forces and moments registered.

θ_2	-10, 0, 10, 20, 30	deg
θ_1	0, 30, 60, 90	deg
ω_i	0, 400, 800	rad/s
(δ_1, δ_2)	$(-30, \pm 30), (-15, \pm 15), (0, 0),$ $(15, \pm 15), (30, \pm 30)$	deg $\times$ deg

Table 1: Flight variables tested in configuration M.

θ_2	0, 30	deg
θ_1	0, 30, 40, 50, 60, 70, 80, 90	deg
ω_i	0, 400, 800	rad/s
(δ_1, δ_2)	$(-30, \pm 30), (-15, \pm 15), (0, 0),$ $(15, \pm 15), (30, \pm 30)$	deg $\times$ deg

Table 2: Flight variables tested in configuration M'.

3.5 Model parameters identification and validation

The strategy for model identification is based on aerodynamic coefficients $C_b(\alpha, \beta, \delta_i)$ and $C_{m,b}(\alpha, \beta, \delta_i)$ 4th order polynomial curve fitting of the data collected in all experimental trials with no propulsion. Afterwards, in order to validate the model, the resulting aerodynamic interpolated data is used alongside the propulsion model described in section 2.2 to predict forces and moments measured in powered model tests. Theoretical and experimental results are compared in Figure 10.

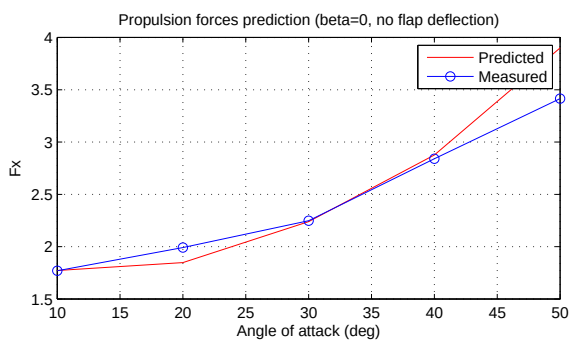


Figure 10: Axial force validation.

The results show accordance between model and measurements for small angles of attack. However, it fails to

predict propulsion components at very high angles of attack. Nevertheless, future work will check whether propulsion model parameters can be adapted to better fit the measurements.

3.6 Winglets study

An outdoor version of the MAVion has been equipped with winglets which act as vertical tails and provide positive stiffness in roll (Figure 12). The triangular shape of the winglets has been designed in order to keep the elevon tips in contact with the winglets even when fully tilted. By doing so, the elevon tips always remain limited by the winglets, which eventually reduces tip losses. As illustrated by Figure 11A which shows the pitching moment gradient with respect to the elevator deflection angle for varying values of the angle of attack, the elevator efficiency is increased when winglets are added. Interestingly, winglets significantly enhance the elevon efficiency particularly in the middle of transition (around 40° angle of attack) when the elevons are tilted at their maximum value (see Fig. 9C). Also, it is noticed that the additional skin friction drag due to the additional winglets wetted area is not visible on the aerodynamic polar (Figure 11B) while the induced drag is significantly decreased at high lift forces. On a practical viewpoint, adding winglets may be useful to protect the elevons and to provide a natural landing device.

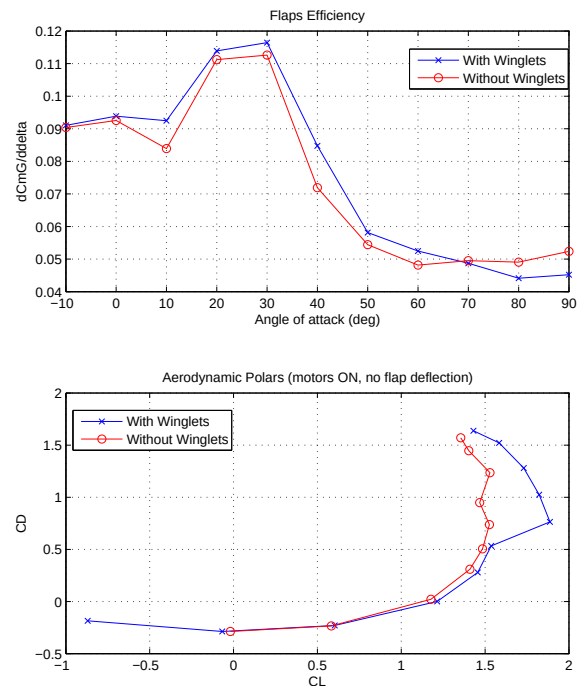


Figure 11: Winglets study.

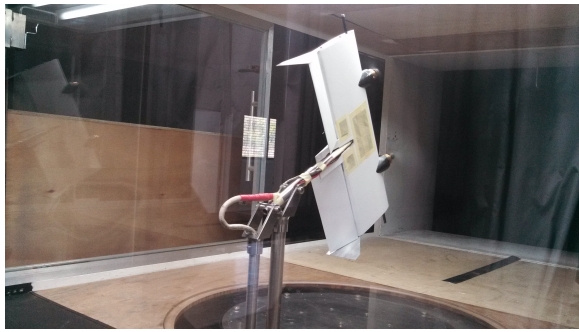


Figure 12: MAVion with winglets version in SabRe.

4 CONCLUSION

In this paper a flight dynamics model for a tilt-body MAV was described. After an overview of the aerodynamics principles that govern the proposed model, a complete wind tunnel campaign is presented. The materials and the set-up procedure for the campaign were described for the MAVion. The equilibrium transition was performed and studied, and validated the MAV design. Additionally, an analysis on the effect of winglets over the overall aerodynamics performance of MAVion is carried out. Finally, the identified model will be used as basis for the flight control systems design of a flying MAVion platform that is currently being manufactured at ISAE.

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] J. Holsten, T. Ostermann, and D. Moormann. Design and wind tunnel tests of a tiltwing uav. *CEAS Aeronautical Journal*, 2(1-4):69–79, 2011.
- [2] S. Shkarayev, B. Bataille, and J.M. Moschetta. Aerodynamic design of micro air vehicles for vertical flight. *Journal of Aircraft*, 45(5):1715–1724, 2008.
- [3] R.H. Stone. Aerodynamic modeling of the wingpropeller interaction for a tail-sitter unmanned air vehicle. *Journal of Aircraft*, 45(1):198–210, 2008.
- [4] 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):229–245, 2011.
- [5] C. Stahlhut and J. Leishman. Aerodynamic design optimization of proprotors for convertible rotor concepts. *Proceedings of the 68th American Helicopter Society International Annual Forum*, pages 592–616, 2012.
- [6] D. Hunsaker and D. Snyder. A lifting-line approach to estimating propeller/wing interactions. *AIAA paper 2006-3466, 24th Applied Aerodynamics Conference*, 5 - 8 June 2006, San Francisco, California, 2006.
- [7] B.J. Gamble and M.F. Reeder. Experimental analysis of propellerwing interactions for a micro air vehicle. *Journal of Aircraft*, 46(1):65–73, 2009.
- [8] M. P. M. a. L. S. Miller. Propeller performance measurement for low reynolds. *Wichita State University, Wichita, Kansas*, 2006.

Deformation Analysis of a Hummingbird Inspired MAV Flapping Wing using Digital Image Correlation

David Kumar*, Akash Ravi Sharma, Tigmanshu Goyal, Vemuri Shyam, P M Mohite, Sudhir Kamle
Department of Aerospace Engineering, Indian Institute of Technology Kanpur, India 208016

ABSTRACT

Flapping wing micro air vehicles (MAVs) have the advantage of being able to fly at slow speeds and have high maneuverability. The design of a flapping wing MAV inspired by birds presents many technical challenges because birds as natural fliers exhibit higher efficiency than any man made flapping wing structure. Thus, the structure and motion of bird wings provide a starting point for the study of flapping wing MAV. In order to compare the deflections of the fabricated wings with the complex mechanism of the bird wings, it is required to quantify these deflections. Both the bird wings and the fabricated wings are very light weight and, therefore extremely light weight sensors or non-contact methods are required for the measurements of the deflections. Digital image correlation (DIC) provides one such option. DIC is an optical non-contact method used to acquire displacements. A simple two dimensional DIC code has been developed and validated in this work. Design and fabrication of wings is based on the hummingbirds. A 3D tapered wing design was developed and used for development of mold. This mold was used for fabrication of flexible polypropylene wings. For flapping motion measurements of wings, a simple slider-crank mechanism was designed to generate an oscillatory motion using brushless DC motors and Arduino Board. For modal analysis, an electrodynamic shaker was used to vibrate the wing with patterns stuck over its surface at its natural frequencies and captured the motion using a high speed camera. The captured images were analyzed using the developed DIC code and mode shapes for 1st and 2nd modes were obtained. For capturing the motion of a wing, a high speed camera mounted on a suitably designed stand was used. The validation of modal analysis was done using commercial finite element analysis software Ansys.

Keywords: MAVs, Digital Image Correlation, Hummingbirds, Modal Analysis.

1 INTRODUCTION

MAVs, small unmanned aircraft, can be used for surveillance, remote sensing, terrain mapping or reconnaissance operations. While they can be fixed wing, or rotary wing, flapping wing MAVs have the advantage of being able to fly at low speeds and have high maneuverability. The development of a flapping wing MAV presents many technical challenges. Some challenges include materials and structural aspects, low Reynolds number aerodynamics, weight and volume constraints, high propulsive power to weight ratio requirements and ability to adapt to all environments.

Nature serves as an effective tool to study how MAVs should be operated. Natural Birds, the original masters of flight, use flapping wings to produce lift and thrust. They exhibit higher efficiency than any man made flapping wing structure. Thus, the structure and motion of their wings provide a starting point for the study of flapping wing MAV. Most birds are able to control their shoulder, elbow and wrist joints. Birds like hummingbird, however, can control only their shoulder joint but with more freedom and variety of motion providing them with the ability to fly in any direction and also hover, thus, making them highly maneuverable. The wing motion of birds can be broken down into flapping, pitching, lead lag and span reduction. Smaller birds use a combination of flapping, pitching and lead-lag while larger birds exhibit mostly flapping and span reduction. A combination of these motions along with the inherent flexibility and structure of the wings gives the birds the ability to generate lift and thrust. The flexibility of wings increases the aerodynamic efficiency of MAVs [1, 2]. Ansari et al [3] studied the motion of insect wing in hover and showed that the motion is in the shape of the figure of eight with the leading edge always leading i.e. the wing pitches so that the leading edge is always in forward motion. In order to understand the complex mechanism of the bird wings, it is required to quantify the deflections of the fabricated wings. The wings of smaller birds, like hummingbird, are very light weight and this prevents the placement of any heavy sensor on them as it will disrupt the weight distribution and add additional weight, which would be large in comparison to original weight of the wing. Extremely light weight sensors or non-contact methods are more effective in measuring the deflections. Digital Image Correlation provides one such option, it is an image registration technique used to measure deformations in two and three dimensions. It uses a series of images captured over a period of time to track and locate the displacement in desired features over the se-

*Email address(es): david@iitk.ac.in

quence. Peters and Ranson [4] presented the technique for implementation in experimental stress analysis by measuring displacement of subsets of deformed surface images using intensity distributions. Sutton et al [5] gave an improvement to the method by suggesting a correlation coefficient based on minimizing the square of the difference between the chosen subset and all similar size subsets in the original array. Luo et al [6] extended the two dimensional DIC to three dimensions by using a pair of cameras along with a pin hole camera model and stereo imaging equations for transformation relating 3D world co-ordinates to 2D image co-ordinates. Tay et al [7] employed the pin hole camera model to determine the apparent change in in-plane displacement due to out of plane rotation and translation. The method however works for rigid motions only. Sutton et al [8] also studied the effect of out of plane motion on in-plane displacements in both 2D and stereovision 3D DIC, and compared the two along with the effects of using a telecentric lens. Even more recently Zhou and Chen [9] used the Zero Mean Normalized Sum of Squared Differences (ZNSSD) instead of the traditional Zero Mean Normalized Cross Correlation (ZNCC) for feature matching. Chen et al [10] used multiple camera system to obtain complete 3D information using stereovision system and the concept of point residual determination for the system of cameras. However, dependency on system setup, stability and calibration still remained in doubt. DIC was also used for MAV flapping wing analysis by Svanberg [11].

The objective of this work is to develop a DIC code that can be used to measure both in plane, out of plane deformations and mode shapes of the developed bio-inspired wing using only one camera. We try to obtain out of plane data by using only one camera along with co-ordinate transformation and a pin hole camera model. Along with the Correlation Coefficient, Fourier transform has been used to speed up the process of analyzing the acquired images in effort to take steps towards achieving live DIC processing. For the purpose of testing the developed code on flexible body motion, the hummingbird is chosen as the bio-inspiration for wing development and polypropylene is chosen as the wing material.

2 DIGITAL IMAGE CORRELATION TECHNIQUE

DIC is an optical non-contact method used to acquire displacements/strains of moving/deforming objects by tracking a unique distribution of pixel intensities of a grayscale image called as pattern or template. These patterns are correlated over a series of images of the object in concern to give the pixel displacement of the surface in the two dimensional images which can then be interpreted according to the three dimensional motion. It is therefore, highly useful when the object in consideration is not accessible, too small, too light, has variation in temperatures or is working at high speeds which prevent the use of traditional contact methods. Commonly, 2D DIC is used to measure in plane displacements. These are helpful in stress calculations and widely used in

experimental mechanics such as to study fractures. Generally two or more than two cameras are used in combination along with stereovision system principles in order to acquire three dimensional locations of points. Errors due to calibration of several cameras to function simultaneously, high precision required for the setup and the cost of obtaining several expensive high speed cameras for dynamic applications are some of its drawbacks.

2.1 Method Development

DIC uses the Correlation Coefficient, or the Correlation Coefficient with Fourier Transform which relies on tracking grayscale intensity values of image pixels in small neighborhoods called subsets or patterns.

2.1.1 Correlation Coefficient

Correlation, given by Equation 1, is the normalized form of covariance. It is a dimensionless quantity and can be used to obtain relative strengths of relations between variables. Mathematically, for n observations of variables x and y ,

$$Corr(x, y) = \frac{\sum_{i=1}^n (x - x')(y - y')}{\sqrt{\sum_{i=1}^n (x - x')^2 \sum_{i=1}^n (y - y')^2}} \quad (1)$$

Where, x' and y' are the means of x and y respectively.

The base of DIC technique lies with the Cross Correlation term. For any program, an input image is read as a matrix, where each pixel contributes to a value in the matrix. The value of this pixel comes from its grayscale intensity in the image. An image is hence, a 2D matrix of intensity values for the purpose of analysis. If $f(x, y)$ is the base image (intensity function at x and y pixel locations) and $t(x, y)$ is the feature or template, for displacements (u, v) , the measure of similarity between them is cross correlation (shown in Equation 2).

$$CCorr(u, v) = \sum_{x, y} [f(x, y)t(x - u, y - v)] \quad (2)$$

The cross correlation however, presents several disadvantages in term of using it in DIC. It is sensitive to the value of the image intensity of image and template. This arises a possibility that the correlation between the feature/template and a bright spot may be greater than the correlation with the actual match. In terms of practical application, it leads to high sensitivity to lighting conditions also, for the same reasons. The Normalized Cross Correlation term, given by Equation 3, or the Correlation Coefficient overcomes this difficulties by subtracting the mean of the image/template and by normalizing the image and feature vectors to unit length yielding a cosine like coefficient ranging from $[-1$ to $1]$ with the 1 being the perfect match.

$$NCCorr(u, v) = \frac{\sum_{x, y} [f(x, y) - f'(u, v)][t(x - u, y - v) - t']}{\sqrt{\sum_{x, y} [f(x, y) - f'(u, v)]^2 \sum_{x, y} [t(x - u, y - v) - t']^2}} \quad (3)$$

Where $f'(u, v)$ is the mean intensity function of the image under the template and t' is the mean of the template.

2.1.2 Fast Fourier Transform

Fourier transform is a mathematical transform which provides one to one transform of signals from time domain to frequency domain. The Discrete Time Fourier Transform (DTFT) is Fourier Transform which receives discrete input in the time domain. This is helpful since in real applications, the signals we receive are discrete and do not have infinite points. The Discrete Fourier Transform (DFT), in addition to discrete time domain, is also discrete in the frequency domain. In 2D Fourier Transform of a image function $f(x,y)$, it is broken down into a linear combination of harmonic (sine, cosine etc) functions. The 2D Fourier can be computed line by line using 1D transform. The outcome is a function of complex variables. The Fast Fourier Transform is an efficient way to compute the Discrete Fourier Transform. The idea behind FFT is that a DFT of length N can be broken down into two DFTs of length $N/2$, first one consisting of odd samples and the second even samples. This means that any N length DFT can be broken down again and again till we get $N/2$ number of 2 length DFTs. For $N = 2^v$ this can be performed $v = \lceil \log_2 N \rceil$ times. Thus the total numbers of multiplications reduce to $N/2(\lceil \log_2 N \rceil)$ thereby giving a complexity of $O(n \log(n))$. As it has been established, that the frequency domain computation of the correlation term is faster in most cases, it would also be required to compute the normalization term or the denominator in the normalized cross correlation term (Equation 3) in frequency domain. This term can be computed easily using the method specified by JP Lewis [12]. It is computed using running sums of entire image first hand and then use them to calculate the mean and energy.

2.2 DIC Code Development and Validation

The DIC technique discussed was developed with the help of MATLAB software and was used to acquire back the applied known deviations and displacements. The purpose of these simulations was to help develop the code step by step, checking and verifying its accuracy and results along the process, before applying it to an experimental setup and finally concluding its feasibility for real world application. During simulations, firstly, one dimensional signals with and without noise were simulated and then extended to two dimensional matrices and images. Finally, a three dimensional rigid body simulation of a wing was performed. Here 3D rigid flapping wing simulation will be discussed.

2.2.1 Pattern Development

Before going forward with 3D simulation of wing motions, it was essential to develop patterns with distinct intensity distributions which could be easily distinguished and identified by the DIC code. These patterns would be placed on the surface of the simulated wing and later printed and pasted on the actual wing. As the wing undergoes motion, the captured

images show an apparent movement of the surface points in two dimension. These patterns are placed on the surface and the DIC code used to capture these apparent two dimensional motions which are then interpreted in three dimensions. MATLAB reads image in the form of a grayscale matrix. Patterns of size 3x3 pixels were developed for use in simulation whereas the size was slightly larger for use in actual experimentation, 6x6 pixels (approximately 2mm x 2mm) to account for the blurring that would occur while printing and recapturing with a camera. The patterns were created with the help of commercial software Adobe Photoshop. Here are some examples of the patterns used. The variations in the



Figure 1: Sample patterns (3x3 pixels).

intensities form a unique pattern which can be matched by the correlation coefficient. For grayscale images, they vary from 0 for black to 255 for white and in between for shades of grey.

2.2.2 3D Simulations

The motion of the bird wing generally consists of three major components, i.e. Flapping, Pitching and Lead Lag. Larger birds also exhibit reduction in span. Here, we try to simulate these three motion and recapture them through a series of images. The model of the wing used is same as that designed while manufacturing a bio-mimicking hummingbird wing. The Gambit model was exported to Autodesk 3ds Max. Autodesk 3ds Max is a 3D computer graphics software primarily used for realistic 3D animation and modeling. We use it to simulate simple rigid rotations about the three coordinate axis, with the patterns used for DIC recognition pasted onto its surface, and render the resulting animation into a series of images to be used in the developed code. The Figure 2 shows the typical rendered image of wing, with two different intensity patterns (x1 and x2) of the size 3x3 pixels, from the simulation upon which DIC code was run. The rendered image was of the size 2048x1556 but that

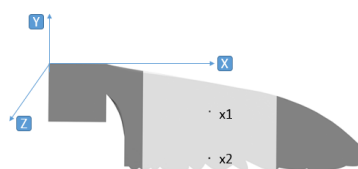


Figure 2: Sample 3D wing model with patterns (top view).

included the extents of the rendered viewport (here, top view

of the wing, as required for DIC), so the effective resolution covering the wing was lesser. Another point of note is that this rendering is not by a camera, i.e. this is not a camera image. There is no perspective projection here. It can be seen as a case of orthographic projection.

2.2.3 Simulation Results

The combined flapping, pitching and lead lag motion corresponds to the combination of out of plane and in-plane displacements i.e. motion about three different axis. Known excitation was given and the interpolated motion curve obtained from the 3ds Max software. The results, from DIC, were then compared with the simulation. Here, due to coupling of motion, one motion may affect the other. We used affine transformation equations to separate out individual components, as we obtained only the net displacement which is a sum of the displacements caused by all the motions. The Figure 3 shows the comparison of DIC results with the software generated motion curves at finite instants of the motion, called frames. The difference in the angles obtained from 3ds Max and the DIC code at each frame is the error.

For lead-lag or in-plane motion the results were found to lie

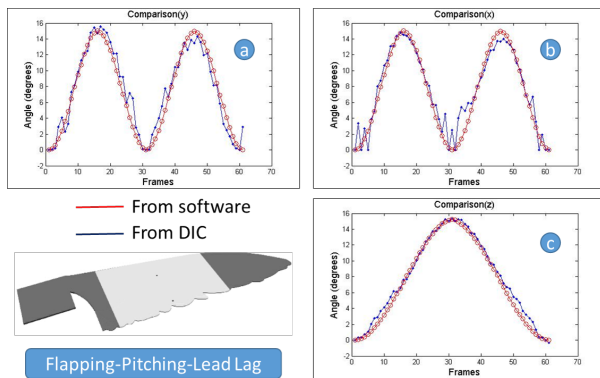


Figure 3: 3D wing simulation: (a) Flapping, (b) Pitching, (c) Lead Lag

in a low error range of 2%-5%, while for flapping and pitching cases the curves were found to deviate from actual response specially at lower angle values. The error obtained for the low angle regime lied in the range of 10%-25%. The simulations were also done for separate cases of flapping, pitching, lead lag. The errors for combined case are higher than in individual motion simulations because of the reflection of error in one term to another, but essentially the curves were replicated.

3 WING DEVELOPMENT AND EXPERIMENTAL SETUP

In this section we discuss the development procedure of a bio-inspired hummingbird wing. Polypropylene, a thermo-

plastic semi rigid polymer, is chosen as the wing material because along with light weight it has good fatigue resistance, flexibility, mechanical strength, toughness, heat and chemical resistance, etc. Developed DIC code is used with dynamic testing experiments of fabricated wings. A high speed camera with a developed flapping mechanism as well as an electrodynamic shaker is used for performing experiments.

3.1 Wing Design and Fabrication

For the development of a successful flapping wing MAV, it is essential that the wings be light, able to flap at high flapping frequencies and amplitude, structurally strong and aerodynamically efficient. In order to make wings with such characteristics we choose hummingbirds as bio-inspiration. The dimensions and other characteristics of the giant hummingbird are taken as the starting point for designing the wing. The Table 1 shows the characteristics of Giant Hummingbird.

The wing was developed using commercial software Gam-

Parameter	Giant Hummingbird
Body Mass (<i>gm</i>)	18-24
Wing Length (<i>mm</i>)	~135
Wing Chord (<i>mm</i>)	~45
Flapping Frequency (<i>Hz</i>)	10-15
Top speed (<i>km/h</i>)	~43.44

Table 1: Giant hummingbird characteristics

bit, using a matching wing structure to the actual wing. Using a detailed and accurate image of the wing, the dimensions are accurately recorded with a digitizer. The same data points were used to replicate the structure in Gambit. The wing design was validated using the plots and equations providing relations between wing length, area and bird weight by Greenewalt [13]. The length of designed wing is 113.8 *mm*. The total wing surface area calculated using Greenewalt Equation 4, for current wing length as reference, is 7222 *mm*² which is close to current total wing area 6932 *mm*².

$$L = 1.0537S^{0.5556} \quad (4)$$

where L and S are in cm and *cm*² respectively. The designed wing sketch was extended to a both side tapered simple solid model. The thickness was varied from shoulder (1.2 *mm*) to tip (0.5 *mm*).

The repeatability is one of the key factors of manufacturing techniques. One must be able to manufacture the same object with same properties repeatedly. Here, we need at least two wings for MAV application and with same characteristics to avoid uncontrolled asymmetric flapping. A mold was designed to cast the desired wing. The mold for the polypropylene wing was made out of brass from the 3-Axis CNC machine. For fabrication of wings the mold, with the material within it, was put in the thermal chamber of UTM in compression mode. The temperature of chamber was set up to 230-

240 degC. The molds are compressed till they completely fit as designed. Final fabricated wing is shown in Figure 4.

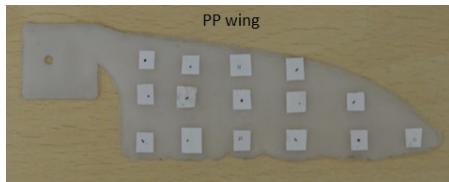


Figure 4: Polypropylene wing with patterns for DIC analysis.

3.2 Experimental Setup

The developed DIC code is used to measure both in plane & out of plane deformations and mode shapes of the developed bio-inspired wing using only one camera. Out of plane displacement data was obtained using co-ordinate transformation. Fourier transform has been used along with Correlation Coefficient to speed up the process of analyzing the acquired images. A high speed camera, Olympus i-SPEED TR, is used for capturing images of the moving wing. It supports a maximum resolution of 1280x1024 pixels till 2000 *fps*. Since it is required to capture the surface of the wing, an appropriate stand was designed to hold the camera safely and in a stable position. DIC requires a speckle pattern as it correlates the high contrast pixel intensities from one image (reference image) to the other image (template image). A black and white speckled pattern is required for a proper correlation. The patterns were attached on the surface of the wing, as discussed in 2.2.1. These patterns were used to depict the displacements of their corresponding sections of the wing.

3.2.1 Wing Motion Measurement Setup

For measurements of wing motion, a motor assembly was designed to provide a flapping motion to the attached wing. The designing process was carried out in commercial software SolidWorks. Each part of the mechanism was separately modeled and assembled. The manufacturing was carried out using FDM Rapid Prototype machine. The manufactured flapping mechanism with high speed camera and setup is shown in the Figure 5.

3.2.2 Modal Analysis Setups

For modal analysis of wings, the electromagnetic shaker is used to vibrate the wing. The natural frequencies were obtained by observing the deflections of vibrating wing through stroboscope and laser displacement sensor. The stroboscope visually shows the nature of mode shapes. For measuring nature of mode shapes using DIC, for 1st and 2nd modes, high speed camera used for recording frames of vibrating wing.

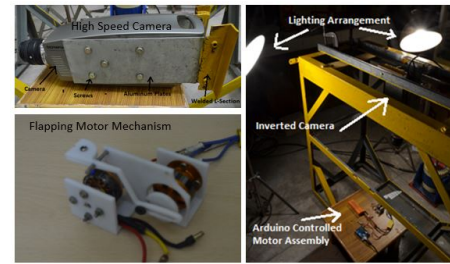


Figure 5: Flapping wing motion measurement setup.

The set-up is shown in Figure 6.

Two 500 Watt bulbs were used as light sources for both the

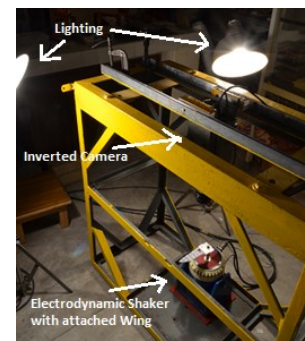


Figure 6: Modal analysis setup.

analysis.

3.3 Modal Analysis Results and Validation using Ansys

The natural frequency and mode shapes of wing were determined with the help of stroboscope and laser displacement sensor. The fundamental natural frequency mode comes at 19 *Hz* and second mode at 78 *Hz*.

To obtain the modal frequencies and mode shapes using Ansys, a three dimensional model of the wing was analyzed using the Modal Analysis module. First the material data for PP was entered, and then solid three dimensional model, made using Gambit software was imported. The dimensions of the model were the same as that of the manufactured polypropylene wing. The model was given a fixed support at its shoulder joint. Meshing was done using patch conforming method with tetrahedrons elements (Figure 7 (a)). The input material

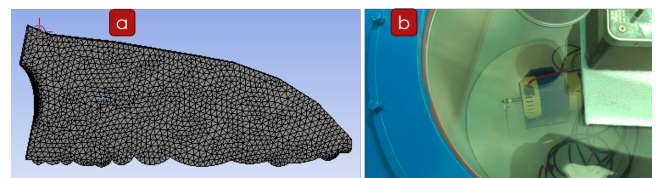


Figure 7: Wing modal analysis: (a) Meshed wing, (b) Wing in vacuum chamber

data (Density = 901 kg/m^3 , Modulus = 1.16 GPa , Poisson's ratio = 0.45) except Poisson's ratio, given for PP wing analysis, was obtained through PP film development and testing. The free un-damped vibration case was taken for solving the problem. In actual case the wing is subjected to two types of damping. One due to system or viscoelastic nature of wing material and second due to fluid or air interaction. A vacuum chamber was designed, developed and used for wing testing (Figure 7 (b)). Using vacuum chamber the contribution of fluid damping (due to air resistance) can be eliminated. Therefore, for validation of experiments, the results from Ansys are compared with vacuum testing results.

4 RESULTS AND DISCUSSION

Here, the results of the dynamic testing of the developed polypropylene wings are discussed.

4.1 Flapping Motion Measurements Results

We use flapping mechanism which was run by the Arduino micro-controller board and capture the motion via the high speed camera at 750fps with f/8 aperture settings. The angles were noted along 4 points on the wing span, point 4 being nearest to the shoulder and point 1 nearest to the tip of the wing. Their respective distances from the shoulder being 10 mm, 35 mm, 60 mm and 85 mm for point 4, 3, 2 and 1. As this can be seen from Figure 8, at point 4 which is nearest to the shoulder, the deflections are lesser as compared to the one farther away because of the bending of the wing. Maximum out of plane motion at point 4 is approximately 21.6° , at point 3 it is 24.7° , at point 2 it is 30.8° and farthest away, at point 1, 34.6° .

The clump of zeros between transition from out of plane to in

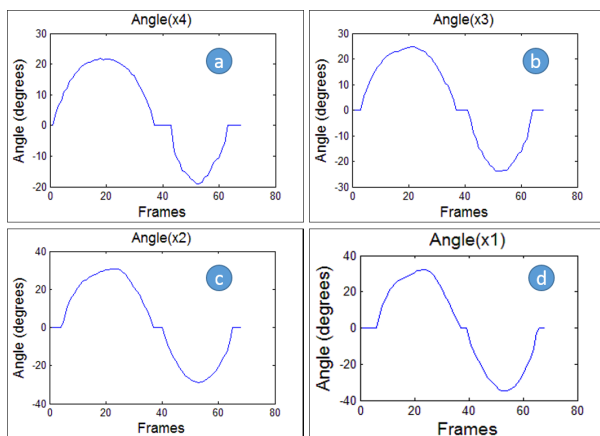


Figure 8: Angles versus Frames at (a) point 4 (b) point 3 (c) point 2 (d) point 1

plane represent the points where angles are small, here taken to be zero. It can also be seen that the point nearest to the shoulder, i.e., point 4 has the largest span of the small angle values which are not captured as the speed along this point

is slower. As we go farther down the span, the speed of the point increases and the number of frames in the range of the small angles which are not interpreted decreases. Also, there is a lag in the cycle with the farther points lagging slightly behind the points nearer to the shoulder of the wing and hence the axis of rotation because of bending.

4.2 Experimental and Computational Modal Analysis Results

The results of modal analysis done in Ansys and vacuum (Table 2) do not match because of two reasons, one is that the un-damped vibration case was solved for the wing and second, may be due to experimental wing development and testing errors. Damping characteristics of wing material, PP, have not been obtained in the present study. The dynamic characterization of wing material using dynamic mechanical analysis can be done to obtain the damping characteristics. The damping characteristics of wing material can be given in Ansys for solving damped vibration case which will decrease the difference between computational and experimental results.

Modes	in Air	in Vacuum	Ansys	Mode shape
1 st	19	19	17.2	Bending
2 nd	78	78	70	Coupled
3 rd	124	124	94.5	Torsional

Table 2: Modal analysis results: Natural frequencies.

Another point to note here is that, from Table 2, the natural frequency values are same from both vacuum and normal atmospheric testing conditions. This is because of a very small surface area and relatively stiff structure of wing for any significant air interaction to occur. Hence the damping due to air does not affect the modal frequencies of this particular wing. So it can be concluded that the aerodynamic loading or air interaction, for the wing designed in present study, does not affect the mode frequencies (at least for first three measured modes). For a more flexible wing in comparison to current wing the natural frequency and wing deflection should increase. Chakravarty [14] did modal analysis on hyperelastic membrane. He found that the natural frequency, with very small change, and deflections increase in vacuum. In present study the mode shapes have not been measured in vacuum. To separate the forces generated by wing during flapping can be done by measuring the forces using force balance in air and vacuum followed by subtraction of forces in vacuum from total force in air.

4.3 Mode Shapes Measured using DIC

For first fundamental frequency or resonance mode, the patterns were stuck along the span of the polypropylene wing to note the displacements along the span. They were placed at known distances in two separate lines as shown in Figure 9

(a), line 1 and line 2.

The setup was run at 19 Hz at suitable qualitative ampli-

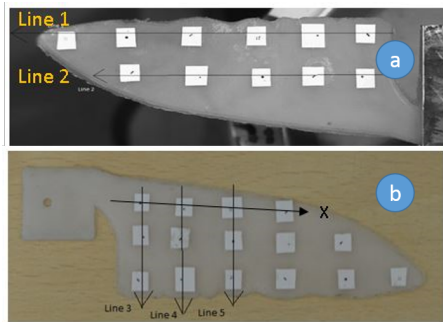


Figure 9: Lines on the wing: (a) Lines along the span of wing, (b) Lines along span and chord of wing

tude and recorded via the high speed camera at 200 fps with aperture set at f/8. At 200 fps, 19 Hz corresponds to 10.52 frames per cycle. The Figure 10 shows that the mode shapes

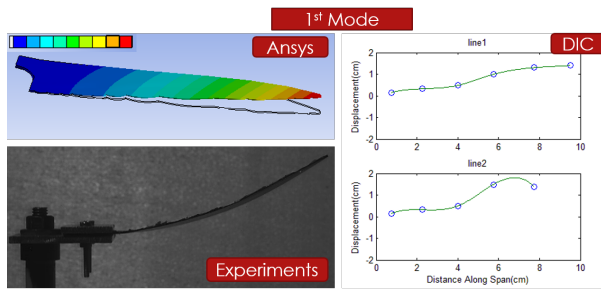


Figure 10: 1st mode

from Ansys, DIC and experiments are similar (where line 1 and line 2 for DIC results corresponds to the lines shown in Figure 9 (a)).

For second mode a third line X, shown in Figure 9 (b), of patterns were placed to provide an additional point to get the twisting nature of coupled mode. The setup was run at 78 Hz (second mode frequency) and vibrating wing motion was recorded at 1000 fps. The nature of mode shapes are compared, Figure 11. Mode shapes are similar which validate DIC code and Ansys results. This mode is coupled in nature because the tail part deflections are higher than leading edge deflections of wing.

4.4 Wing-tip Displacement Response

The objective of this study is to determine the relation between wing-tip response and wing-root excitation. The wing was attached to the electromagnetic shaker and excited at different root amplitudes for first resonance mode. The displacements of wing-tip and wing-root were measured using laser displacement sensor with NI-DAQ cards. The Equation 5 shows the proportionality between wing-tip response and

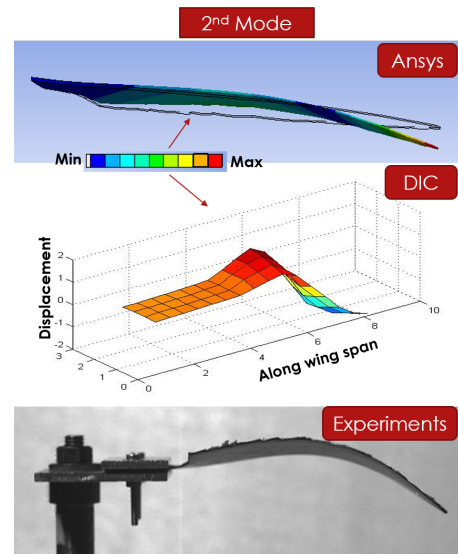


Figure 11: 2nd mode

wing-root excitation as obtained from Figure 12 (d).

$$Y_{tip} = 13.1 * Y_{root} \quad (5)$$

where Y_{tip} , Y_{root} are wing-tip and wing-root displacements respectively. The experimental setup and results for this analysis are shown in Figure 12.

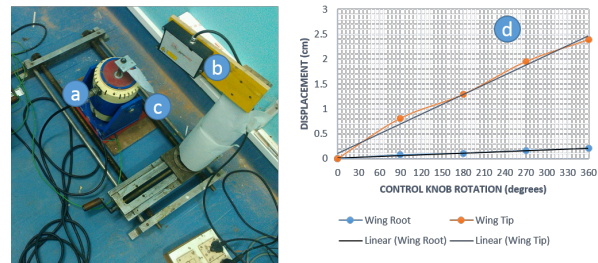


Figure 12: Wing-tip response: (a) Electromagnetic shaker, (b) Laser displacement sensor, (c) Wing, (d) Results

5 CONCLUSION

The aim of this study was to develop a non-contact method to study the fabricated hummingbird inspired wing deformations. DIC was chosen as the suitable method. Firstly, the fundamentals of Correlation and Frequency domain analysis using Fourier Transform were established. A DIC code was developed with the help of MATLAB based on capturing a designed template with unique grayscale intensity distribution. Simulations were carried out to validate the developed code. Patterns of size 3x3 pixels were developed for use in rigid body simulations. Rigid body simulation of a hummingbird

wing was carried out. High errors were consistently seen for small angles of rotation less than 2° . However, they reduced drastically for angles larger than 2° and stayed less than 5 % for nearly all cases. The results were better for in plane motions than out of plane motions. Error was also seen to be dependent on proximity with the axis of rotation. For the combined motions, errors obtained were higher because of the error in one term being reflected in the other.

For experimental application, DIC code was used for measurements of flexible body motion, design and fabrication of wings based on the Hummingbird. The wings were made of polypropylene. A simple slider-crank mechanism was designed to generate an oscillatory motion using brushless DC motors and Arduino Board. For capturing the motion, a high speed camera mounted on a suitably designed stand was used. The captured images were analyzed using the developed DIC code and the flapping motion was obtained. An electromagnetic shaker was used to vibrate the wing. The first and second modes were determined using stroboscope and laser displacement sensor. The three dimensional model of the wing was analyzed through Ansys modal analysis to validate the experimental results. The natural frequencies values from Ansys were compared with experimental modal analysis values of wing in vacuum chamber. The high speed camera was used to record the vibrating motion of wing (with patterns). The stored images were analyzed using DIC code for determining the mode shapes which were compared with experimental and Ansys mode shapes. The mode shapes (from DIC, Ansys and Experiments) were found to be similar. It was therefore found that the developed DIC code was a reliable method to analyze flapping wing deformations, which were then studied to mimic the Hummingbird wing motions with a reasonable accuracy.

ACKNOWLEDGEMENTS

We would like to acknowledge the support of the following departments from IIT Kanpur: ME (4i lab), AE (workshop and structures lab) and MSE. We also thank NP-MICAV, India.

REFERENCES

- [1] S. Heathcote, Z. Wang, and I. Gursul. Effect of span-wise flexibility on flapping wing propulsion. *Journal of Fluids and Structures*, 24(2):183 – 199, 2008.
- [2] Y. WenQing, S. BiFeng, S. WenPing, and W. LiGuang. The effects of span-wise and chord-wise flexibility on the aerodynamic performance of micro flapping-wing. *Chinese Science Bulletin*, 57(22):2887–2897, 2012.
- [3] S.A. Ansari, R. bikowski, and K. Knowles. Aerodynamic modelling of insect-like flapping flight for micro air vehicles. *Progress in Aerospace Sciences*, 42(2):129 – 172, 2006.
- [4] W.H. Peters and W.F. Ranson. Digital imaging techniques in experimental stress analysis. *Optical Engineering*, 21(3):213427–213427, 1982.
- [5] M.A. Sutton, W.J. Wolters, W.H. Peters, W.F. Ranson, and S.R. McNeill. Determination of displacements using an improved digital correlation method. *Image and Vision Computing*, 1(3):133 – 139, 1983.
- [6] P. F. Luo, Y.J. Chao, M.A. Sutton, and W.H. PetersIII. Accurate measurement of three-dimensional deformations in deformable and rigid bodies using computer vision. *Experimental Mechanics*, 33(2):123–132, 1992.
- [7] C.J. Tay, C. Quan, Y.H. Huang, and Y. Fu. Digital image correlation for whole field out-of-plane displacement measurement using a single camera. *Optics Communications*, 251(13):23 – 36, 2005.
- [8] M.A. Sutton, J.H. Yan, V. Tiwari, H.W. Schreier, and J.J. Orteu. The effect of out-of-plane motion on 2d and 3d digital image correlation measurements. *Optics and Lasers in Engineering*, 46(10):746–757, 2008.
- [9] Y. Zhou and Y.Q. Chen. Feature matching for automated and reliable initialization in three-dimensional digital image correlation. *Optics and Lasers in Engineering*, 51(3):213–223, 2013.
- [10] F. Chen, X. Chen, X. Xie, X. Feng, and L. Yang. Full-field 3d measurement using multi-camera digital image correlation system. *Optics and Lasers in Engineering*, 51(9):1044–1052, 2013.
- [11] C.E. Svanberg. *Biomimetic micro air vehicle testing development and small scale flapping wing analysis*. Air Force Institute of Technology, Wright Patterson Air Force Base, Ohio, Ohio, 2008.
- [12] J.P. Lewis. Fast normalized cross-correlation, 1995.
- [13] C.H. Greenewalt. *The flight of birds : the significant dimensions, their departure from the requirements for dimensional similarity, and the effect on flight aerodynamics of that departure*. American Philosophical Society, Philadelphia, 1975.
- [14] U.K. Chakravarty. Analytical and finite element modal analysis of a hyperelastic membrane for micro air vehicle wings. *Journal of Vibration and Acoustics*, 135(5):051004, 2013.

V-REP & ROS Testbed for Design, Test, and Tuning of a Quadrotor Vision Based Fuzzy Control System for Autonomous Landing

Miguel A. Olivares-Mendez*, Somasundar Kannan, and Holger Voos

Automation Research Group

Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg

ABSTRACT

This paper focuses on the use of the Virtual Robotics Experimental Platform (V-REP) and the Robotics Operative System (ROS) working in parallel for design, test, and tuning of a vision based control system to command an Unmanned Aerial Vehicle (UAV). Here, is presented how to configure the V-REP, and ROS to work in parallel, and how to use the developed packages in ROS for the pose estimation based on vision and for the design and use of a fuzzy logic control system. It is also shown in this paper a novel vision based fuzzy control approach for the autonomous landing task on a static and on a moving platform. The control system is based on four fuzzy logic controllers (FLC) working in parallel on an external control loop based on the visual information. All the controllers were designed and tuned to command the vertical, longitudinal, lateral, and heading velocities of the UAV.

1 INTRODUCTION

Simulation environments are always an important tool in research, even more in the specific field of robotics. For this research topic there are many of them, and each one has its one limitations. The Webots [1] is one of the most used, but there is no free version for academia and the cost is not that cheap. The toolbox of robotics for Matlab done by Peter Corke [2] is recommendable to understand many robotics control problems and its vision based on outer control loop. The disadvantage of this approach is that is difficult to make any modification to adapt the environment to each user necessities because of its innumerable lines of code. Another disadvantage of this approach is that there is not direct way to use what is implemented in the toolbox with a real robot. This is something that was solved with ROS [3]. The Robotic Operative System provides an open source framework to develop packages to interactive with real sensors, actuators, robots, etc, using a publisher/subscriber system for the communi-

cation between them. The main advantage of this pseudo-operative system, is that is easy to use, to develop new packages and to communicate with existing packages. The big number of users and developers in the ROS community make this software more interesting indeed. This framework comes with a 3D simulation environment called Gazebo 3D [4]. The main advantage point of this software is that all the packages (algorithms, control, etc) used in the virtual world of Gazebo, can be used with minor changes in the real version of the simulated robot. This fact implies an enormous reduction of time in the software implementation part of any research. The disadvantage of the Gazebo simulator is the high requirements of CPU power and graphic card. The next step in 3D environment simulators is the software developed by Coppelia Robotics, the Virtual Robotics Experimentation Platform (V-REP) [5]. In comparison with the Gazebo software, this software can be installed and run without a powerful graphic card and does not required a powerful CPU. The V-REP comes with a large number of robots, sensors and actuators models, and several structures to create a virtual world just dragging and dropping. This simulator allows to interact with the virtual environment during the simulation running time. It is very easy to check how the control system response against disturbances, position changes of the target location or the addition of more robots, objects, structures or sensors in the scene. Another advantage of this software is the bridge with ROS [6], allowing to use everything developed in the previously mentioned framework. All these characteristics make the V-REP and the connection with ROS the ideal platforms for learn, teach, research and developed with robots.

This work focuses on the vision based control system of a quadrotor in the simulated environment to be used later with a real aircraft. A specific task of autonomous landing on a moving target with a quadrotor has been defined to test the V-REP and ROS connection, how the V-REP experimental platform works with quadrotors, and how the developed ROS packages works. There are many visual servoing applications present on the literature. Different vision-based algorithms have been used to follow a car from a UAV [7], [8],[9]. Visual terrain following (TF) methods have been developed for a Vertical Take Of and Landing (VTOL) UAV [10]. In [11] a description of a vision-based algorithm to follow and land on a moving platform and other related tasks are proposed. A cooper-

*Corresponding author. E-mail address: miguel.olivaresmendez@uni.lu; Tel.: +352 46 66 44 5478

active strategy has been presented in [12] for multiple UAVs to pursue a moving target in an adversarial environment. The low-altitude road following problem for UAV using computer vision technology was addressed in [13]. People following method with Parallel Tracking and Mapping (PTAM) algorithm has been developed in [14]. The autonomous landing approach presented in this work is based on the control of the lateral, longitudinal, vertical, and heading velocities of the quadrotor to modify its position to land on a predefined platform.

The outline of this paper is structured in the following way. Section 2 presents the configuration of the V-REP experimental platform. Section 3 shows the specific vision algorithm used for the autonomous landing task. The fuzzy control approach and the developed ROS package for this purpose are explained in section 4. Section 5 shows the experiments done with the developed testbed to tune the controls system approach and to test it for the autonomous landing on a static, and on a moving platform. Section 6 shows the conclusions and the future work.

2 V-REP SIMULATION ENVIRONMENT CONFIGURATION

In this section is presented the mayor details of the simulation environment of V-REP, as well as the modifications done, and the connects with ROS.

The V-REP presents an easy and intuitive environment to create your own virtual world and to include any of the robots that are provided, as well as objects, structures, actuators and sensors. Also, it allows to create your own robot by adding actuators, joints, sensors and basic forms. An example of a V-REP environment is shown in Figure 7. On the left side of the environment there is a list of robots, sensors, actuators, structures, etc that could be easily include in the simulation scene by drag and drop (model browser). In the next column, there is the scene hierarchy, where all the robots, sensors, graphs and structures of the current scene are represented. A script based on *LUA* script could be associated to each sensor and robot to interact with them, inside the V-REP environment of from the outside (e.g. C++ code or from a ROS package). The central window could be divided in one or more views, we divided it in four views, two external cameras (top and back), and the representation of the velocities and the position of the UAV. More detailed information about this robotics experimental platform is found in [5].

The V-REP comes with a quadrotor model that is configured to follow position commands. When a vision sensor is used to control a robot the most common way is to use velocity commands. In this work is presented a vision based control system, also known as Image Based Visual Servoing (IBVS), in which the control commands have to be velocity commands. The existing QR model presents some stability problems when the velocity commands are sent, therefore, some modifications needed to be done on the model based on

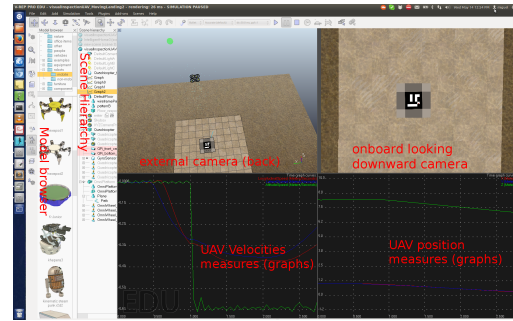


Figure 1: Capture frame of the V-REP environment.

the assumption that the symmetric inertia matrix is a diagonal matrix such as $\text{diag}(I_{xx}, I_{yy}, I_{zz})$, as is presented in [15]. Basically a very low value of I_{xx} introduces poor damping in the roll attitude so purely a velocity control loop does not suppress the faster modes. So the quadrotor is drifting sideways. This is very common in real experimental quadrotors. In such cases it is always needed an outer position loop to hold the position of the quadrotor.

Once, the model was modified the inner control loop has to be adapted to the new model. Four classical PD controllers are defined for the hovering controller of the heading, roll, pitch and height. In this case it is taken into account the current measures of aircraft's angles (roll, pitch and heading), the altitude estimation, and the aircraft's velocities (longitudinal, lateral, heading and altitude). The inner controllers developed in this phase for the virtual quadrotor were tested in a non exhaustive way. It is checked the correct behavior of the quadrotor hovering and against soft disturbances. The authors really recommend this tool and this process for teaching purpose, because of the high sensibility of the simulator environment and the new quadrotor model against minors changes in the controllers parameters.

The next step is to have the onboard camera feedback available to be processed in a ROS environment. The virtual quadrotor comes with two cameras onboard, but their image feedback can not be sent or published to be used by ROS. For this purpose it is necessary to add one (or more, depends on your needs) new vision sensor and attach it to the quadrotor. In the added vision sensor is possible to modify the child script to share the current image capture by the vision sensor to be processed by the corresponding ROS package. Following the ROS policy to exchange information, the image is published in a ROS' topic.

The specific visual algorithm developed as a ROS package has to get the current frame published, process it and send the extracted information to the control system. All this process is done by different developed ROS packages that will be explained into details in the next sections. There is also another process to be accomplish in the V-REP. The virtual quadrotor must receive the control commands and apply

them. It is also done in a child script, but in this case, in the one that is associated with the virtual quadrotor. As well as in the image sharing process, in this case we have to define a ROS' type subscriber to get the control feedback as a new velocity reference for the inner control system.

All the implemented code explained in this section and the model adaptation of the quadrotor to receive velocity commands, as well as a scene example to use them are available on [16]. Most of them are based on the information extracted from the forum web site of Coppelia software [5].

3 VISION ALGORITHM

The vision algorithm had to obtain the position of the landing platform. With this information, the control system approach has to be able to command the UAV to center landing platform in the image, orientate it, and approximate it to the landing platform. The vision algorithm is not the principal purpose of this work, for this reason we use a visual algorithm based on the detection, recognition, and processing of augmented reality (AR) markers or codes. The idea is to have a moving landing platform with a code printed on it. Based on a markers database, the camera calibration parameters, and the size of the code, the algorithm is able to estimate the pose of the camera (quadrotor) respect to the marker, that is the orientation of the camera (the quadrotor) in the three axis, the distance between the camera (the quadrotor) and the code, as well as the lateral and vertical displacement versus the center of the marker. The developed ROS package for the visual algorithm is the adaptation to ROS of the ArUco software [17], a developed C++ library of augmented reality that uses OpenCV. It is an improved Hamming Code based algorithm with an error detection.

The ArUco-ROS package, called *aruco_eye* is subscribed (that is the ROS method to get information shared by other packages) to the specific image streaming publisher from the camera (real or virtual). The current frame is processed and the extracted information is sent by a publisher defined for this purpose.

To use this package with the virtual camera and a real one it is needed to include the camera calibration and the specific topic's name in the location where the camera published the images. This could be done by the command line or configuring a roslaunch file. To do the tests in the virtual environment a panel with one of the ArUco codes added as a texture is included, as it is shown in the Figure 2. In the real world the codes were printed and paste into a wall.

The image processing algorithm consists on the estimation of the distance between the ArUco target code and the UAV in the three axis (longitudinal, lateral and vertical distances), as it is shown in the Figure 3.

The developed ROS package presented in this section and some examples of how to use it, are available on [16].

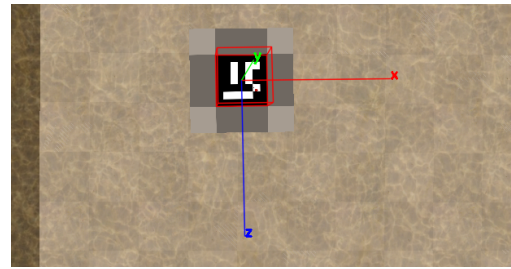


Figure 2: Virtual image captured by the virtual camera on the UAV and processed using the ArUco ROS package.

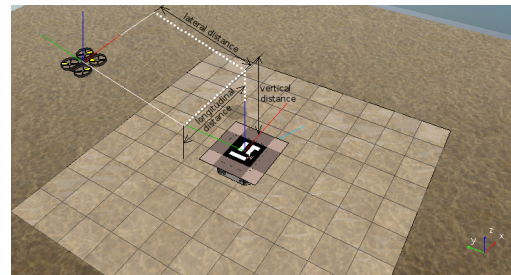


Figure 3: Explanation of the image processing algorithm.

4 FUZZY CONTROL SYSTEM

The autonomous landing task defined in this work is based on the capability of the quadrotor to change its position to land over the moving landing platform. The UAV must center the landing platform, and once it is centered start to descend. To solve this task a control system approach was designed using four fuzzy controllers working in parallel. The longitudinal and lateral speed controllers keep the UAV positioning to have the moving landing platform in the center of the image. The vertical speed controller approximates the UAV to the landing platform. The heading controller modify the heading of the UAV to have the landing platform well oriented. The longitudinal, lateral, and heading velocity controllers have been designed as a fuzzy PID-like controller, with three inputs and one output. The vertical speed controller is just a simple fuzzy PD-like controller. The longitudinal, lateral, and vertical speed controllers have as outputs velocity commands in meters per seconds, and for the heading velocity controller is degrees per seconds. Figure 4 shows the control loop of the control system approach implemented for this work.

All the controllers were defined in a simply way, with just three sets per each input, and five sets for the output, defined using triangular functions, that means that the rules' base is composed just with 27 rules for the longitudinal, lateral and heading velocity controllers and 9 rules for the vertical speed controller. The defuzzification method used is the height weight and the inference motor is the product. The

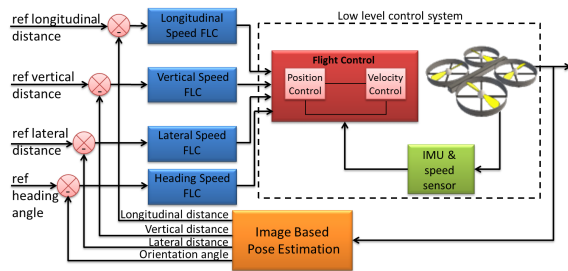


Figure 4: Control loop of the presented approach.

rule base was defined based on the heuristic information of the relation of the three inputs. The Figure 5 shows the basic design of the controllers before any modifications. In the longitudinal and lateral speed controllers were taken into account the current pitch and roll angles of the UAV in the information obtained by the vision algorithm for the estimation of the translation in x and y axis respected to the moving landing platform.

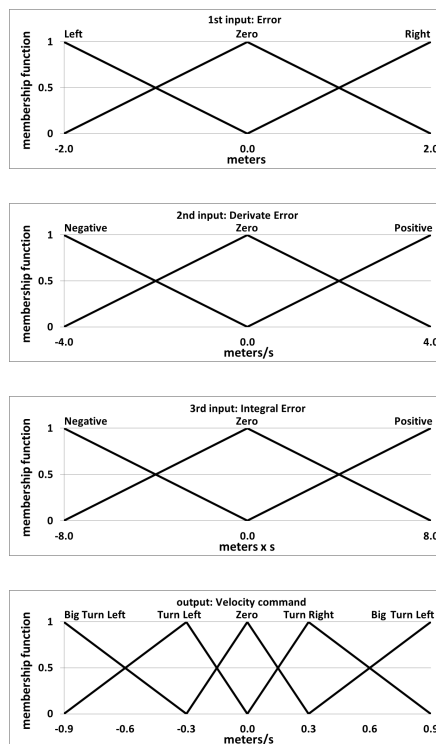


Figure 5: Initial design of the inputs and output variables for the fuzzy logic controller.

The Tables 1, 2, 3 show the initial definition of the rule base.

To implement the fuzzy controllers in the ROS environment a new ROS package was developed called MOFS-ROS.

Dot/error	Left	Zero	Right
Negative	Left	Zero	Right
Zero	Zero	Right	Right
Positive	Right	Right	Big Right

Table 1: Base of rules with value for the third input (integral of the error) equal to **negative**, before the manual tuning process

Dot/error	Left	Zero	Right
Negative	Left	Zero	Zero
Zero	Left	Zero	Right
Positive	Zero	Zero	Right

Table 2: Base of rules with value for the third input (integral of the error) equal to **zero**, before the manual tuning process

Dot/error	Left	Zero	Right
Negative	Big Left	Left	Left
Zero	Left	Left	Zero
Positive	Left	Zero	Right

Table 3: Base of rules with value for the third input (integral of the error) equal to **positive**, before the manual tuning process

This ROS package is the adaptation to ROS of the own developed C++ library MOFS (Miguel Olivares' Fuzzy Software) [18]. As well as the C++ library, this new ROS package (MOFS-ROS) allows to implement fuzzy controllers in an easy way, loading the specific characteristics of the controller and the rule base from two different txt files. This package interacts with the roscore and other packages by a service, that provides a new output each time that new inputs were received by the common subscriber/publisher ROS' communication policy. The specification of the subscriber and the publisher is done by a parameter in the *roslaunch* command line or by a *roslaunch* file.

This ROS package allows to create new fuzzy controllers using triangular or trapezoidal membership functions, the product or the maximum inference motor and the height weight defuzzification method. Extended possibilities will be included in the next versions of this ROS package. More detailed information of this software is found in [19] in where is explained the C++ library version.

An extra ROS package was developed to put across the information from the visual algorithm, to the fuzzy control system. This package is also in charge of sending the controllers' output to the virtual or real quadrotor. This package receives two different names, when it is used with the virtual quadrotor is called *VREP_VC* (VREP visual control), and (visual_control_system). This package is the only one that has to be modified depending the UAV to be used, the number of controllers to use, and the error's signals to control. In this way the ArUco-ROS package and the MOFS-ROS package is kept without significant changes to be used with the virtual or the real environment, or either for future control approaches. This package also contains some extra process

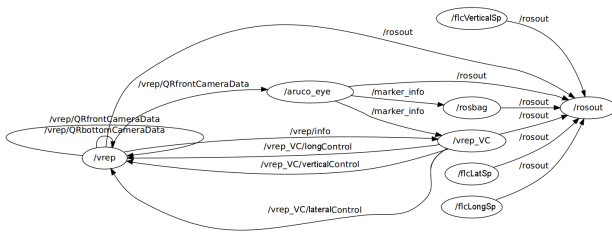


Figure 6: Interaction between all the active processes from V-REP and ROS during the simulator tests.

when a real AR.Drone quadrotor is in use. This process is called *emergency_ardrone_control*, and it allows to send basic commands to take off, land, and to do an emergency stop to the AR.Drone. Through this process the user can also select the specific ArUco code to set as a target.

All the developed ROS packages presented in this section and some examples of how to use it are available on [16].

5 EXPERIMENTS

The experimental part is divided in two phases. In the first one, the controllers have been tested and tuned using a step signal to keep one code in the center of the image in the V-REP environment and its connection with ROS. After that the tuned controllers were tested all together for the autonomous landing task.

5.1 Tuning Process

The simulation environment was set by a quadrotor with a looking forward camera, and a panel with one ArUco code, as it is shown in Figure 7. The location of the quadrotor is set to have a two meters step signal when it starts to work.

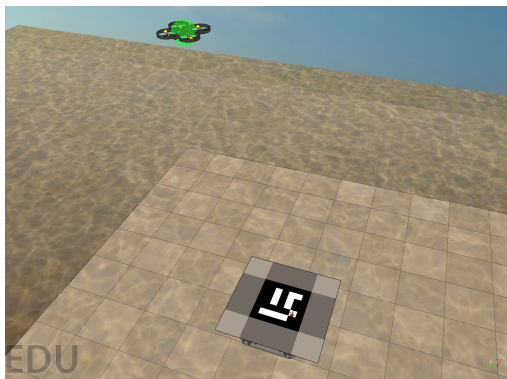


Figure 7: V-REP environment design with a quadrotor and the ArUco code.

A complete schema of all the processes involved in the simulation environment is shown in Figure 7. Where *vrep* is the V-REP simulator environment, explained in section 2, *aruco_eye* is the process of the vision algorithm explained in section 3, the *fllatSp*, *fllongSp*, and *fllverticalSp* are

the fuzzy controllers for the lateral, longitudinal and vertical speed respectively, and *fllOrientSp* (explained in section 4). The *vrep_VC* is the process that shares the visual information to the control system and send the control commands to the virtual quadrotor in the V-REP (explained at the end of the section 4). The *rosbag* is an internal ROS package to store some data of each test. Finally, the *rosout* is the core of the ROS system.

During the manual tuning process, the range of the variables are first adapted and then the output of some specific rules. The V-REP simulation environment and in conjunction with ROS and the developed ROS' packages allow to test the controllers and to modify the different characteristics of the controllers easily. Because of the big similarities between the lateral, longitudinal, and heading controllers, the tuning process is applied to one of the three controllers, and then the results are used in the others. The tuning process is defined by the response of the lateral speed controller responding against a 2 meters step signal. First, the inputs' range have to be adjusted. The Figure 8 shows the response of some of the different configuration tested for the range of the inputs. It is started with the initial controller explained in section 4, represented as *MF1* with the blue line in Figure 8. Then several modifications are tested on the range of the inputs and the output, some of them are shown in the mentioned Figure as *MF2*, *MF3*, *MF4*, being the last one (represented by the green line) the one that gets a better response. The variables definition of the three controllers is shown in Figure 9.

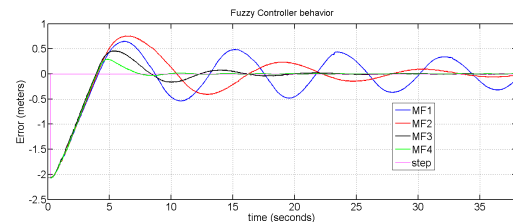


Figure 8: Response of the different lateral speed controllers during the tuning phase of the inputs' range adjustment.

Based on this controller, tuning process it is continued with the adaptation of the rules' base. In this phase the output of few rules were modified to reduce the overshoot presented in the response of the best controller obtained in the previous phase (the *MF4*). Also for this phase the test is the same than in the previous phase, a step signal of 2 meters. The Figure 10 shows the behavior of some of the controllers tested. The behavior of the different controllers tested are quite similar unless one of them, that reduces completely the overshoot presented in the others (*Rules4*). The Tables 4, 5, 6 show the final state of the rule base after the manual tuning process. From the initial base of rules presented in section 4, 10 output's rules were changed.

Once the lateral speed controller is tuned, this information

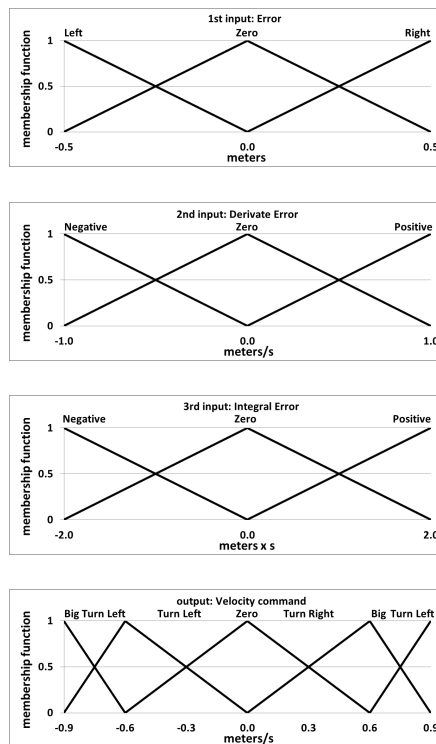


Figure 9: Final design of the variables of the fuzzy controller after the manual tuning process.

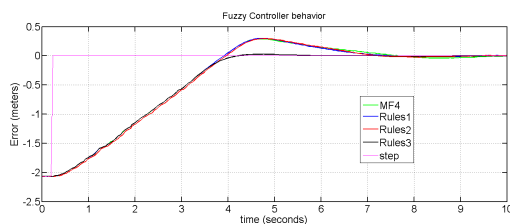


Figure 10: Response of the different lateral speed controllers during the tuning phase of the rules' base adaptation.

Dot/error	Left	Zero	Right
Negative	Big Left	Left	Right
Zero	Left	Zero	Right
Positive	Right	Right	Big Right

Table 4: Base of rules with value for the third input (integral of the error) equal to **negative**, after the manual tuning process

Dot/error	Left	Zero	Right
Negative	Left	Left	Zero
Zero	Left	Zero	Right
Positive	Zero	Right	Right

Table 5: Base of rules with value for the third input (integral of the error) equal to **zero**, after the manual tuning process

Dot/error	Left	Zero	Right
Negative	Big Left	Left	Left
Zero	Left	Zero	Right
Positive	Left	Right	Big Right

Table 6: Base of rules with value for the third input (integral of the error) equal to **positive**, after the manual tuning process

is used to set the other controllers, the longitudinal, vertical, and heading velocities. An adaptation to degrees (inputs), and degrees per seconds (output) was done for the heading controller.

5.2 Autonomous Landing Tests

Once the four controllers are obtained, they can be used all together to control the virtual quadrotor in the specific task of autonomous landing. Two different test are defined. In the first one, the landing platform is static, and in the second one it change its position in the plane x, y and turn over the z axis. In both cases the code is located over a ground robot, to be easy to configure the movements of the landing platform. The initial position of the quadrotor is the same in both cases, 8.0 meters of altitude, and a displacement of 1.6 and 0.9 meters respect to the landing platform for the x, y axis of the quadrotor respectively. The vertical velocity controller has the constraint of not to act until the error in the x and y axis of the QR are reduced under 0.5 meters. This is an strategy to reduce the potential disturbances created by the action of the descend of the Quadrotor.

The next Figures shows the behavior of the control system and the quadrotor for the autonomous landing on a static platform. The Figure 11 shows the evolution of the error for the lateral and longitudinal velocities controllers. This Figure shows how the error was reduced to zero in less than 8 seconds. The Figure 12 shows the evolution of the error for the heading controller.

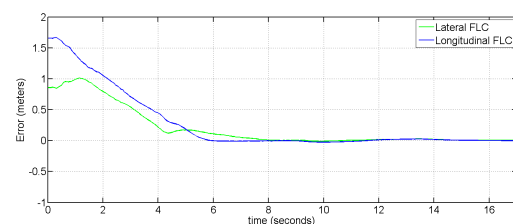


Figure 11: Evolution of the error for the longitudinal and lateral velocities controllers for the autonomous landing test on a static platform.

The Figure 13 shows the evolution of the position of the quadrotor in the three axis x, y, z during this test. This Figure also shows how the vertical velocity controller keep without any action until the error in x and y axis are reduced. Finally, the Figure 14 shows the evolution of the absolute heading of

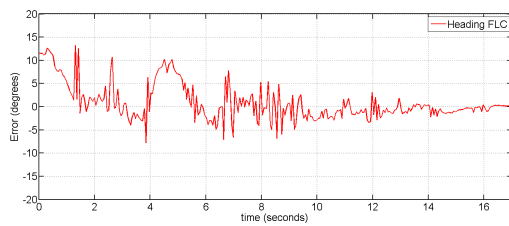


Figure 12: Evolution of the error for the heading velocity controller for the autonomous landing test on a static platform.

the quadrotor, and how the heading is affected by the action of the other controllers.

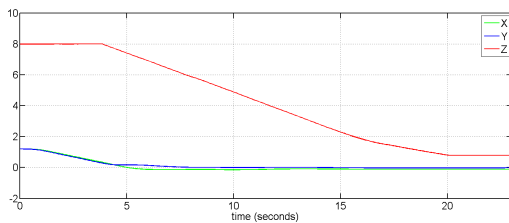


Figure 13: Evolution of the quadrotor position in x,y,z axis in the autonomous landing test on a static platform.

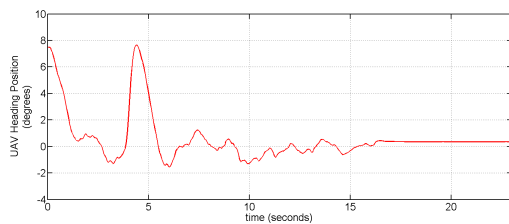


Figure 14: Evolution of the absolute heading of the quadrotor in the autonomous landing test on a static platform.

After the test of the autonomous landing on a static platform, a test on a moving platform was defined. Because the code was set on an omni-directional ground robot (available in the V-REP environment), it is possible to configure different movements and rotations for the landing platform. A set of movements in all direction was set for this test, and also a rotation in z . The evolution of the error for the lateral and longitudinal velocities controllers are shown in Figure 15. The Figure 16 shows the evolution of the error for the heading velocity controller.

The evolution of the quadrotor position is shown in Figure 17. The Figure 18 shows the evolution of the absolute heading of the quadrotor.

The behavior of the control system approach was evaluated using the root mean square error (RMSE) during both tests. The Table 7 shows the RMSE values for the longitudinal, lateral and heading velocities controllers.

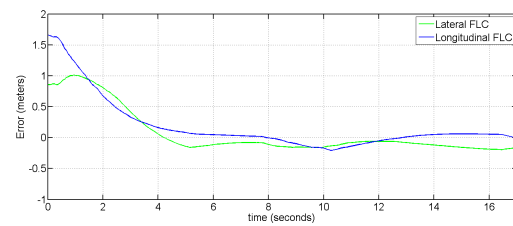


Figure 15: Evolution of the error for the longitudinal and lateral velocities controllers for the autonomous landing test on a moving platform.

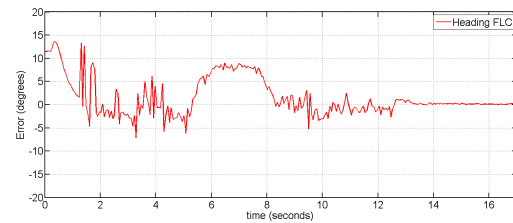


Figure 16: Evolution of the error for the heading velocity controller for the autonomous landing test on a moving platform.

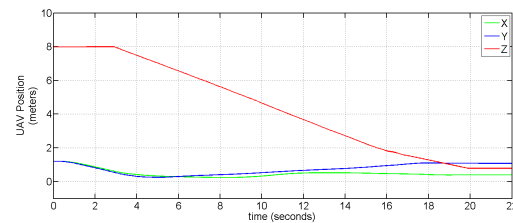


Figure 17: Evolution of the quadrotor position in x,y,z axis in the autonomous landing test on a moving platform.

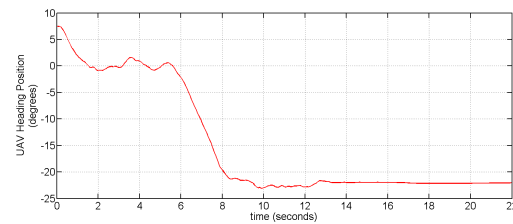


Figure 18: Evolution of the absolute heading of the quadrotor in the autonomous landing test on a moving platform.

Controller type	Static Platform (RMSE value)	Moving Platform (RMSE value)	units
Longitudinal	0.5346	0.4615	meters
Lateral	0.3661	0.3777	meters
Heading	3.9029	4.3728	degrees

Table 7: Root mean square error for the Longitudinal, Lateral and Heading velocities controllers in the two tests presented.

The videos related to the tests presented in this section are available in [20].

6 CONCLUSIONS AND FUTURE WORK

In this paper is presented the process a testbed developed using the V-REP & ROS frameworks to design, test and tune a vision based control system to command an aircraft. A ROS package was developed to design, and use fuzzy controllers in ROs and specifically with a quadcopter. A C++ library for the detection of augmented reality codes was adapted to be used in ROS, and to estimate the relative position of the quadrotor respect to the code. Another ROS package to communicate the visual algorithm and the fuzzy control approach was also developed. The principal idea of this package is to make easy the uses of other visual algorithms and control approaches. It is also presented in this work a novel fuzzy control approach to command a quadrotor for autonomous landing tasks using the testbed, and all the developed ROS packages. The control system approach was tune in the simulator using a step signal, to be tested later in a more complex tasks as the autonomous landing on a static and on a moving platform. The good results obtained were evaluated by the RMSE with values below 40 cm for the lateral velocity controller, below 55 cm for the longitudinal velocity controller, and below 4.5 degrees for the heading controller. The author are focus now in test the control system approach with a real quadrotor for the autonomous landing task.

REFERENCES

- [1] Webots official site. <http://www.cyberbotics.com>, 2014.
- [2] Peter corke's robotics toolbox for matlab. http://petercorke.com/Robotics_Toolbox.html, 2014.
- [3] Robot operating system (ros). <http://ros.org>, 2014.
- [4] Gazebo 3d simulator. <http://ros.org/wiki/gazebo>, 2014.
- [5] Coppelia robotics. virtual robotics experimentation platform (v-rep). <http://www.vrep.org>, 2014.
- [6] Wiki site of the ros bridge with v-rep. http://wiki.ros.org/vrep_ros_bridge, 2014.
- [7] P. Campoy, J. F. Correa, I. Mondragón, C. Martínez, M. A. Olivares, L. Mejías, and J. Artieda. Computer vision onboard uavs for civilian tasks. *Journal of Intelligent and Robotic Systems*, pages 105–135, 2009.
- [8] W. Ding, Z. Gong, S. Xie, and H. Zou. Real-time vision-based object tracking from a moving platform in the air. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 681–685, 2006.
- [9] Celine Teuliere, Laurent Eck, and Eric Marchand. Chasing a moving target from a flying uav. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS2011*, pages 4929–4934, 2011.
- [10] F. Ruffier and N. Franceschini. Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, pages 2339–2346, 2004.
- [11] D. Lee, T. Ryan, and H.J. Kim. Autonomous landing of a vtol uav on a moving platform using image-based visual servoing. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 971–976, 2012.
- [12] U. Zengin and A. Dogan. Cooperative target pursuit by multiple uavs in an adversarial environment. *Robotics and Autonomous Systems*, pages 1049–1059, 2011.
- [13] Joseph Egbert and Randal W. Beard. Low-altitude road following using strap-down cameras on miniature air vehicles. *Mechatronics*, pages 831–843, 2011.
- [14] G. Rodríguez-Canosa, S. Thomas, J. del Cerro, A. Barrientos, and B. MacDonald. A real-time method to detect and track moving objects (datmo) from unmanned aerial vehicles (uavs) using a single camera. *Remote Sensing*, pages 1090–1111, 2012.
- [15] G. Antonelli, F. Arrichiello, S. Chiaverini, and P.R. Giordano. Adaptive trajectory tracking for quadrotor mavs in presence of parameter uncertainties and external disturbances. In *Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on*, pages 1337–1342, July 2013.
- [16] Miguel angel olivares-mendez snt homepage, 2014.
- [17] Aruco: a minimal library for augmented reality applications: [online]. <http://www.uco.es/investiga/grupos/ava/node/26>, 2013.
- [18] M.A. Olivares-Mendez, P. Campoy, C. Martinez, and I. Mondragon. A pan-tilt camera fuzzy vision controller on an unmanned aerial vehicle. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2879–2884, Oct.
- [19] I. Mondragón, M. A. Olivares-Méndez, P. Campoy, C. Martínez, and L. Mejias. Unmanned aerial vehicles uavs attitude, height, motion estimation and control using visual systems. *Autonomous Robots*, 29:17–34, 2010. 10.1007/s10514-010-9183-2.
- [20] Youtube channel of the automation research group at snt-university of luxembourg: Automation research group snt.uni.lu. https://www.youtube.com/channel/UCBkpapz06ViwK_, 2014.

Development and Application of PP-CNT Composite for Hummingbird Inspired MAV Flapping Wings

Vemuri Shyam Kumar,*David Kumar, Tigmanshu Goyal, P M Mohite, S Kamle
Indian Institute of Technology Kanpur, India-208016

ABSTRACT

Micro Air Vehicles (MAVs) are small unmanned aircrafts which have a maximum size limit of 150 mm in any direction. They can be used for surveillance, reconnaissance, targeting, etc. To perform such missions, MAVs are required to hover. Hummingbirds, having excellent flight characteristics (such as hovering, ability to fly in any direction, ability to produce a reverse camber during upstroke for generating lift in both up-down strokes), have been chosen as the bio-inspiration for wing development. Wings are required to be light, strong and fatigue resistant, to be able to properly flap during flight. Therefore, wing-material becomes a crucial component. An optimization analysis, on the basis of density and fundamental frequency values obtained through Ansys, was done for selecting the wing material. Polypropylene (PP) was observed to have desired properties such as light weight, flexibility, strength, fatigue resistance, good heat and chemical resistance etc. Mixing Carbon Nano Tubes (CNTs) with PP can further increase the strength significantly, making it more suitable for large amplitude flapping. The PP-CNT composites were developed using solution casting method. The films were characterized mechanically (using UTM). The wings were characterized by their structural dynamic properties. The modal analysis of wings was done to obtain natural frequencies and mode shapes. The analysis was aimed to get the fundamental mode in the flapping range (8-15 Hz) of hummingbirds, as resonance increases efficiency. It was also done inside vacuum chamber to observe the effect of air on the natural frequency and modes. The Ansys results were compared with the experiments in vacuum for validation of experimental results. Damping coefficient of wings was also determined. In the end, bio-mimicking of hummingbird wing was also tried by doing some material and structural advancements in the wings.

Keywords: MAVs, Hummingbirds, Bio-mimicking, PP-CNT composites, Structural Dynamic Analysis

1 INTRODUCTION

Micro Air Vehicles (MAVs) are small unmanned aircraft with a maximum size limit, set by DARPA, 150 mm in any direction. MAVs are of three types, namely, fixed wing, rotary wing and flapping wing. The flapping wing MAVs are better than other two, mainly in indoor flights. Fixed wing MAVs cannot hover and its agility is also not good while rotary wing MAVs have wall proximity problem. Also, flapping wings MAVs are able to fly at low speeds or hover for long duration. They generate lift and thrust using their wings only which is advantageous in terms of efficiency. Natural fliers are the original masters of flight. They are the perfect source of inspiration for development of MAVs. Raney and Slominski [1] developed hummingbird inspired mechanism and wings. They used carbon-epoxy composite, for wing frame, and a thin latex, for wing membrane, as the wing materials. Zduinch et al [2] also made similar structure wing using carbon fiber prepreg strips as vein structure and Mylar as membrane. Wood [3] also designed hummingbird inspired wing and used carbon fiber as wing frame with polyester as wing membrane. Ratti and Vachtsevanos [4] took inspiration from flight mechanism features of dragonfly and developed a bio-inspired MAV. Nakata et al [5] developed and did aerodynamic testing of hummingbird inspired MAV.

During flight, the wing experiences loads due to given acceleration or inertial forces and due to air interaction of aerodynamic forces. For flapping wing MAVs, wings are the only source to generate all the required forces to be able fly and perform missions. So wings have to be light, strong and fatigue resistant to make a successful MAV. To make such wings, appropriate materials have to be chosen. Among most commonly used polymers, Polypropylene (PP) which has light weight, good fatigue resistance, flexibility, chemical and heat resistance etc., can be used as the wing material. For further increase in strength, carbon nanotubes (CNTs) have been used. CNTs are most commonly used inorganic fillers for PP. CNTs can be classified into single-walled CNTs (SWCNTs) and multi-walled CNTs (MWCNTs). CNTs have excellent mechanical, thermal properties which makes them an excellent reinforcing material. Their elastic modulus is ~ 1 TPa for SWCNT, $\sim 0.3 - 1$ for MWCNT and tensile

*Email address(es): vemurishyam3@gmail.com

strength is 50 – 500 GPa for SWCNT; 10 – 60 GPa for MWCNT [6]. They are very light weight too, their density is one-sixth of steel which can be as low as 1.3 gm/cm^3 [7]. The aspect ratio of CNTs is also high [6] which is advantageous in terms of better reinforcement of the matrix. The PP-CNTs nanocomposite can be prepared using solution mixing, melt blending and in-situ polymerization. The benefit of using solution mixing is that mixing of nanotubes with matrix in proper solvent, through sonication and stirring processes, improves the dispersion quality of composite [8]. The weight concentration percentage of CNTs with respect to PP is a very important factor. At high concentration, agglomeration of nanotubes occurs, which degrades the properties of the composite [9, 10]. Due to agglomeration, the load from matrix to nanotubes cannot be transferred properly [11].

The wings, during flight, get subjected to the interaction of kinematics, structural dynamics and aerodynamics [12]. Much experimental and computational work has been done in the field of kinematics and aerodynamics. The structural dynamics analysis of wings is very important and has yet to be explored. Sladek et al [13] did aerodynamic and structural dynamic testing of wings, and obtained results for determination of the repeat-ability of their manufacturing technique.

The objective of this study is to make PP-CNTs composites and use them for development of MAVs flapping wings. The performance of nanocomposite will be evaluated by mechanical, physical and thermal characterization methods. The wing will be designed using dimensional parameters of the hummingbird wing. The wing manufacturing technique will also be evaluated on the its repeat-ability. The fabricated wings will be characterized on the basis of their structural dynamic characteristics. The validation of experimental results will be done by doing analysis using commercial finite element analysis software ANSYS. The modal testing of wings was also carried out inside a vacuum chamber. The results from ANSYS and vacuum testing will be compared for validation of results later. At last, bio-mimicking design of the hummingbird wing will be discussed.

2 WING DESIGNING AND MATERIAL DEVELOPMENT

The most of the research related to MAVs is based on kinematics, controls and aerodynamics. To make a successful MAV, the wings have to be designed properly and appropriate materials have to be used for fabricating them. The wings have to be light, flexible, structurally strong and aerodynamically efficient. The flexibility of wings affects the efficiency of MAV. Controlled flexibility along the wing span can be beneficial in terms of thrust production [14].

2.1 Wing Designing

Nature serves as an effective tool to study the way MAVs should be operated. Natural fliers, use flapping wings to produce lift and thrust. Among the numerous examples of highly successful flapping fliers that exist in nature, the Hummingbird is chosen as the biological inspiration. The Hummingbirds, because of their specific flapping mechanism and wings, have excellent characteristics such as long duration hovering, backward motion, high agility, etc. They produce a reverse camber during upstroke to generate lift in both up & down strokes during hovering [1]. These characteristics, taken from various sources, also lie in the range of parameters set by DARPA for MAVs. The comparison of MAVs and hummingbirds is shown in Table 1.

Parameter	MAVs	Hummingbirds
Max size (mm)	150	50-305
Weight (gm)	10-100	2-24
Payload (gm)	1-18	NA
Endurance (min)	20-60	>120
Airspeed (km/h)	35-60	40-54
Range (km)	1-10	>6
Hovering	Hover is trade-off with endurance/range	~Hour

Table 1: Comparison of hummingbirds and MAVs characteristics.

Here, Pantagona Gigas (Giant Hummingbirds) is chosen as the bio-inspiration for development of wings. They are largest hummingbirds among all hummingbird species.

Parameter	Giant Hummingbirds
Body Mass (gm)	18-24
Wing Length (mm)	~135
Wing Chord (mm)	~45
Flapping Frequency (Hz)	10-15
Top speed (km/h)	~43.44

Table 2: Giant hummingbird characteristics.

The wing design development was done using Gambit software. A printed image of actual Giant hummingbird wing was used for recording the dimensions. The profile of wing on the printed image was marked to create guide lines for generating data points. A digitizer was used to generate the data points. Large number of data points were generated for making smooth 2D profile of wing. Generated data points were imported into Gambit for making computational wing sketch. The designed wing sketch is shown in Figure 1.

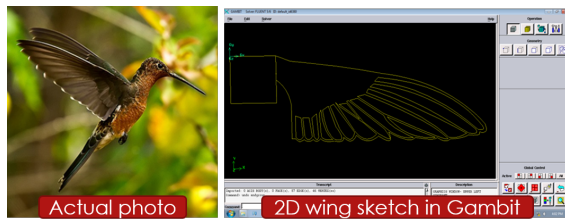


Figure 1: Designed wing sketch.

The Table 3 shows the dimensions of designed wing sketch. The dimensions are in the range of giant hummingbird values which validates the design.

Parameter	Designed Wing
Wing Length (<i>mm</i>)	120.18
Wing Chord (<i>mm</i>)	44.0
Wing Area (<i>mm</i> ²)	7894

Table 3: Designed wing characteristics.

The design has been validated using relations given in Equations 1, 2, 3 between wing length (*L*, *cm*), wing area (*S*, *cm*²) and body weight (*W*, *gm*) determined by Greenewalt [15]. Using wing length of designed wing as a reference parameter, other parameters such as wing area, body weight determined and validated.

$$L = 2.329W^{0.556} \quad (1)$$

$$L = 1.0537S^{0.5556} \quad (2)$$

$$S = 4.1712W \quad (3)$$

Another equation (4) was given by Greenewalt (used by Raney and Slominski [1]) for determining flapping frequency (*f*, *Hz*) these relations for a particular wing length (*L*, *m*).

$$f = L^{-\frac{5}{4}} \quad (4)$$

Using all the above relations, given by Greenewalt, some parameters for a hummingbird based on length of designed wing can be determined. The Table 4 shows the parameters obtained using Equations 1, 2, 3, 4. These parameters fall in the range of giant hummingbirds characteristics values.

Parameter	Designed Hummingbird
Wing Length (<i>mm</i>)	120.18
Body Mass (<i>gm</i>)	19.12
Wing Area (<i>mm</i> ²)	7992
Flapping Frequency (<i>Hz</i>)	14.12

Table 4: Designed hummingbird parameters.

2.2 Wing Material Optimization

The material of wings should be light weight, flexible, fatigue resistance, strong etc. Materials should also have heat resistance for high temperature application and chemical resistance for application in chemically dangerous environment. Hummingbirds, during hovering, flap their wings at high amplitude and frequency. The idea here is to develop a wing which has a fundamental frequency in the flapping frequency range of hummingbirds. We then use the phenomena of resonance, to get a higher amplitude flapping at approximately the same power input. Many insects and birds flap their wings at resonance or fundamental natural frequency of the wings[1].

For selecting wing material, an optimization analysis will be carried out on the basis of material density and the fundamental natural frequency of the wing.

Commonly used polymeric MAV wing materials such as polypropylene (PP), polyethylene (PE), polyvinyl chloride (PVC), polycarbonate (PC), polystyrene (PS), polyvinylidene fluoride (PVDF), acrylonitrile butadiene styrene (ABS) and polytetrafluoroethylene (PTFE) or teflon, polyamide (PA) or nylon, polyethylene terephthalate (PET) or Mylar are chosen as wing material. The essential input material data required for performing modal analysis in ANSYS are density, elastic modulus and Poisson's ratio. The material properties of all the polymers, except PP, were taken for analysis from reliable sources such as MatWeb (<http://www.matweb.com/>), Wikipedia (http://en.wikipedia.org/wiki/Main_Page), Goodfellow (<http://www.goodfellow.com/>) and INEOS (<http://www.ineos.com>). For input data of PP the experimental values were taken from tensile testing of films.

A simple 3D tapered (1.1 mm thickness at root to 0.36 mm thickness at tip) model of wing from designed 2D sketch was developed in Gambit. The analysis parameters except material data were same for all the cases. The wing was meshed into 102416 elements using tetrahedrons. The figure 2 shows the comparison plot of fundamental frequency vs. density for all materials.

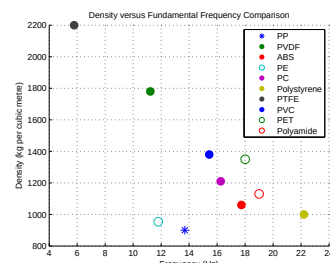


Figure 2: Modal Analysis: Wing material optimization.

As can be seen, the PP and PE data fall in the desirable range. We chose PP over PE because it has better fatigue resistance, lower density, higher softening and melting point, higher rigidity and hardness etc.

2.3 PP-CNT Composite Development

Polypropylene, a thermoplastic semi-rigid polymer, is widely used in a variety of industrial applications including plastic hinges, packaging, reusable containers etc. It is an addition polymer made from Propene and is known for good fatigue resistance, mechanical strength, chemical resistance, toughness, heat resistance etc. A CNT is a tube shaped material made of carbon, having a diameter measuring on nanometer scale (ranging from μ 1 nm up to 50 nm) approximately 1/50000th of the width of human hair and length several microns to centimeters (18 centimeters). Basically, a Carbon Nano Tube (CNT) is a new form of carbon, equivalent to two dimensional graphene sheet rolled into a tube, by configuration. CNTs are known as excellent nano fillers because of their properties such as high tensile strength (60GPa), modulus of elasticity (around 1TPa), light weight and high aspect ratio.

2.3.1 Composite Development Procedure

Materials used for composite development are Polypropylene from Oddo, MWCNTs from Nanoshel and Xylene from Merck.

20ml of xylene is added to 20gms of PP and heated in a beaker to 210° at 60rpm. simultaneously, 0.02gms(0.1wt% of PP) is added to 20ml of xylene and sonicated for 45min. Both the solution are mixed and stir for 30min, till the solution becomes homogeneous. Solution is taken in petri dish and heated in oven at 140° till the solvent gets evaporated. The composite film of PP-CNT is obtained.

For making thinner and uniform films, the final composite sample was put between film mold plates and pressed using UTM in compression mode. The processing temperature was maintained at $230-240^{\circ}\text{C}$. The applying load (around 9 kN) was optimized to get the film thickness below the 0.1-0.2 mm mark. A similar procedure was used for making neat PP films.

2.3.2 Composite Characterization

The developed films were characterized using UTM for mechanical performance.

Mechanical Characterization:

The tensile film testing standard ASTM D1708-13 was used for making specimen. Instron 3345 was used for testing the composites. 5 specimens were tested for each category, at

1 mm/min speed. The Figure 3 shows the tensile testing results. The tensile strength and Young's modulus of composite increased with MWCNTs weight concentration till 0.1 %. After 0.1 wt % there was a continuous decrement in the performance.

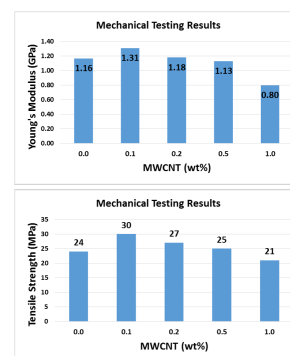


Figure 3: Mechanical results of composites.

3 WING DEVELOPMENT AND TESTING SETUP

In this section, we discuss the development procedure of a Bio Mimetic Hummingbird Wing using polypropylene and PP-CNT composites. Modal coupling of bending and torsion was to be achieved. This is because, it provides flap and passive pitching motions required for controlled and efficient flapping flight as seen from the studies on bird wing motion. The fabricated wings were tested using electro-dynamic shaker for modal analysis.

3.1 Wing Fabrication

Repeat-ability is one of the key factors of manufacturing techniques. One must be able to manufacture the same object with same properties repeatedly. Here, we need at least two wings for MAV application with same characteristics to avoid uncontrolled asymmetric flapping. A mold was designed to cast the desired wing shape. The mold for the polypropylene wing was made out of brass from the 3-Axis CNC Machine at the 4i Laboratory, IIT Kanpur. The wing shape designed in Gambit was extended to a both side tapered solid model. The thickness was varied from root (0.8 mm thick) to tip (0.3 mm thick), to facilitate good bending and utilizing the favorable properties of the polymer. The thickness gradation leads to a variation in the value of flexural stiffness along the wing span from the root to the tips and mimics the variation in a bird wing, qualitatively. The objective is to obtain coupled bending and torsional modes which can represent flapping and passive pitching motion, at this very small size scale, thus simplifying the actuation mechanism which is one of the major causes of weight in the flapping wing MAV.

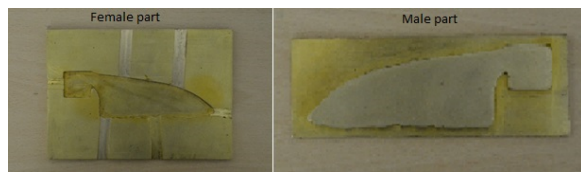


Figure 4: Manufactured Lower and Upper Mold for PP Wing.

For fabrication of the mold, with the wing polymer material in it, was put in the thermal chamber of UTM in compression mode. The temperature of chamber was set up to 230-240 °C. As the temperature increases, the force reading on the machine increases as the material expands. Once it crosses the glass transition temperature and begins to flow the force applied starts reducing. The molds are compressed till they completely fit as designed. The thickness of the wing varied from 1.1 mm at the leading edge of the root to 0.5 mm at the trailing edge, despite the designed mold having variation from 0.8 mm to 0.3 mm because of inaccuracy in manufacturing of mold and the casting process not being perfect because of the low tendency of propylene to flow.

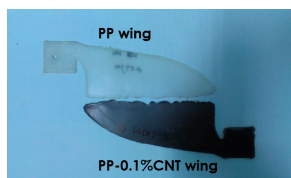


Figure 5: Fabricated wings.

3.2 Wing Testing Setup

An electrodynamic shaker was used to vibrate the wings for performing modal analysis. The natural frequencies and mode shapes were found with the help of Stroboscope and laser displacement sensor. The Figure 6 shows the experimental setup for modal analysis in open air.

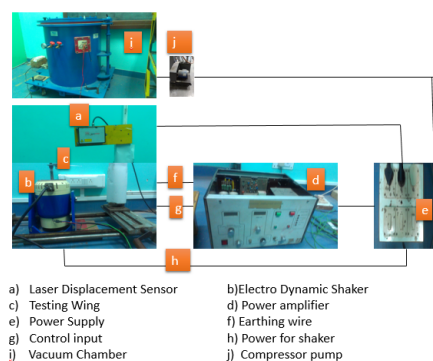


Figure 6: Wing testing setup

For modal analysis in vacuum, same setup is used but the shaker is kept inside the vacuum chamber and using compressor pump vacuum is created. For actuation of wing all the wires (f, g, h in 6) are taken out from the vacuum chamber through the holes (can be seen in vacuum chamber (h) in 6). These holes are packed with adhesives.

3.3 Damping Coefficient Calculation Experiment

Damping study of the hummingbird wings is an important aspect of study because the wing must be able to reduce unwanted vibrations due to sudden wind gusts. This requires an estimate of the logarithmic decrement (δ) and the damping ratio (ξ) of the polypropylene wing. The mathematical relations used for calculating these quantities are given in Equation 5 and 6.

$$\delta = \ln \frac{x_1}{x_2} \quad (5)$$

$$\xi = \frac{1}{\sqrt{1 + \left(\frac{2\pi}{\delta}\right)^2}} \quad (6)$$

Where, x_1 and x_2 are the first and second maximum positive amplitudes. An experimental setup consisting of a Laser Displacement Sensor, a LabVIEW program for data acquisition and a wing clamp was made. The wing was clamped at the root section and excited by a sudden jerk at the tip. Tip displacements were measured via laser and recorded as a function of time to study the damping characteristics. The laser sensor was first calibrated before being used in the damping experiment to obtain a relation (Equation 7) between the laser displacement voltages (V) and the physical displacements (d).

$$d = 29.444 * V - 0.0404 \quad (7)$$

4 RESULTS AND DISCUSSION

Here Modal analysis and damping results will be discussed.

4.1 Experimental Modal Analysis Results

The fundamental frequency is closer to the flapping range of designed wing.

Modes	PP wings (Avg. of 3)	PP-0.1CNT wings (Avg. of 3)	Mode shapes
1st	16	16	Bending
2nd	63	66	Coupled
3rd	95	95	Torsional
4th	146	152	Coupled

Table 5: Modal analysis: Experimental results.

It is observed that resonance frequency remains almost equal with increase in the strength and stiffness.

4.2 Damping Experimental Results

The nature of amplitude variation with respect to time, shown in Figure 7, was observed for the polypropylene wing.

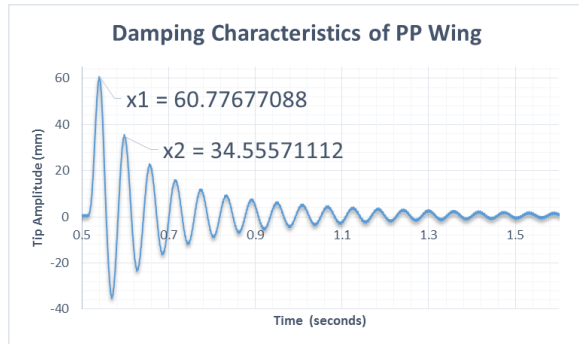


Figure 7: Damping Characteristics Plot

Using the relations mentioned in Equation 5, the logarithmic decrement and the damping ratio were obtained as 0.5646 and 0.0895, respectively.

4.3 Modal Analysis using Ansys

ANSYS 14 was used for modal analysis of designed wings. The 3D model of wing was imported from Gambit to ANSYS Design Modeler. The input material data, given for analysis, was taken from the film testing results, barring the Poisson's ratio data, which was taken from literature as stated earlier. The Table 6 shows the input material data.

Density (kg/m^3)	901
Modulus (GPa)	1.16
Poisson's ratio	0.45

Table 6: Ansys modal analysis: Input material data

The meshing of wings, shown in Figure 8 was done using patch conforming method. Tetrahedron elements were used for meshing the wings. Pre-Stress is not applied and the wing is fixed at the root of the wing.

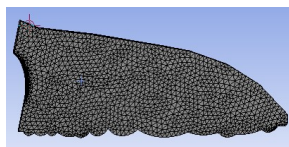


Figure 8: Ansys modal analysis: Meshed wing.

The convergence analysis, shown in Table 7 was done on the basis of number of elements. This analysis is done to

check the developed finite element modal analysis code. Frequency drops, for first mode, at an exponential rate with respect to the number of elements and saturates at 13.68 Hz beyond 200,000 elements, which was obtained by increasing the number elements in steps. Hence, convergence of solution was obtained.

Modes	Natural Frequency (Hz)		
	Elements (1940)	Elements (18990)	Elements (102383)
1st	15.007	14.191	13.768
2nd	64.76	58.536	53.312
3rd	84.73	78.666	74.653
4th	184.39	149.78	125.12

Table 7: Ansys modal analysis: Convergence

4.4 Validation of Experimental Results: Ansys Results vs. Vacuum Results

The free undamped vibration case was considered for solving the problem computationally in ANSYS. However, in the actual case, the wing is subjected to two types of damping, namely the damping due to the system itself and due to the surroundings. The system damping is due to the viscoelastic nature of wing material and the surrounding damping is due to fluid or air interactions. A vacuum chamber was designed, developed and used for wing testing. Using the vacuum chamber, the contribution of fluid damping (due to air resistance) can be eliminated. Therefore, for validation of experiments, the results from ANSYS are compared with vacuum testing result and shown in table 8.

Modes	Vacuum	Ansys	Mode shape
1st	16	13.7	Bending
2nd	62	53.3	Coupled
3rd	96	74.6	Torsional
4th	147	125.1	Coupled

Table 8: Comparison of ANSYS and Vacuum results

4.5 Effect of Air loads

It is observed, there is not much change in value of modal frequencies in air and vacuum. This is because of stiff structure and a very small surface area for any significant air interaction to occur. Hence the damping due to air does not affect the modal frequencies of these particular wings. So it can be concluded that the aerodynamic loading or air interaction, for the wing designed in present study, does not affect the mode frequencies (at least for first three measured modes).

4.6 Nature of Mode Shapes: Experiments vs. Ansys

The mode shapes from Experiments and Ansys are matching.

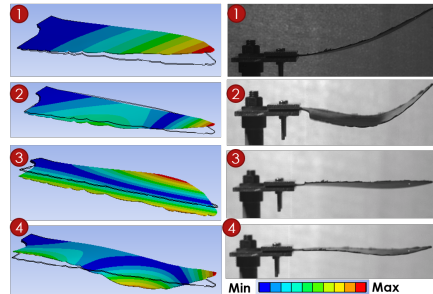


Figure 9: Modal Analysis: Nature of mode shapes.

4.7 Repeatability of Wing Manufacturing Technique

Repeat-ability of wing manufacturing technique is very important. The technique should be able to produce the wings with same characteristics (mass, structural and aerodynamics). Here the repeat-ability is determined on the basis of mass and structural dynamic characteristics. For evaluating the repeatable wing producing capability of manufacturing technique different composite, epoxy-CNT, wings were developed and tested. These wings are shown in the Figure 10. For epoxy-CNT composite case, the 0.2 wt% MWCNTs concentration performed well.

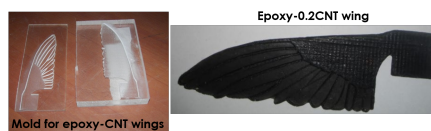


Figure 10: Epoxy-CNT composite wings.

In the Table 9 the small values of standard deviation indicates that the technique is able to produce same mass wings.

Wing type	Average (gm)	Standard Deviation
Neat PP wings	2.74	0.15
Epoxy-0.2CNT wings	6.4	0.26

Table 9: Repeatability by mass characteristics.

The Table 10 also indicates that the manufacturing technique is able to make wings with same structural dynamic characteristics.

Modes	PP wings		Epoxy-0.2CNT wings	
	Average	Standard Deviation	Average	Standard Deviation
1st	16	0.58	40	1.15
2nd	63	2.52	171	5.69
3rd	95	3.51	242	11.59
4th	146	5.13	419	19.43

Table 10: Repeatability by structural dynamic characteristics.

4.7.1 Bio-mimicking of Hummingbird Wing using PP-CNT Composite Membrane

For mimicking of wing frame, carbon fiber epoxy composite is used. Developed PP-0.1%CNT composite film is used as the wing membrane. Same PP wing mold was also used to make this composite structure wing. This wing is also tested for its natural frequencies and mode shapes which are shown in Table 11.



Figure 11: Bio-mimicked hummingbird wing.

Modes	Natural frequency (Hz)
1st	21
2nd	83
3rd	109
4th	126

Table 11: Bio-mimicked wing's experimental results.

5 CONCLUSION

A simple tapered hummingbird inspired wing was designed and validated for its properties. The modal analysis of wings using different materials was done and used for optimizing the material for wing. Polypropylene was chosen over other commonly used polymeric materials on the basis of density and fundamental frequency. The testing of developed composites showed that CNT (at 0.1 wt%) enhanced the stiffness and strength of neat PP. After successful composite development and mechanical testing, the wing fabrication was carried out using a mold. Both neat PP and optimized PP-CNT (0.1 wt%) combination composite were used

for wing fabrication. Modal analysis of wings was done using electromagnetic shaker, stroboscope, laser displacement sensor and high speed camera. The natural frequencies and corresponding mode shapes of the wings were recorded using a high speed camera. The validation of experimental values was done by results from ANSYS. These computational results were compared with the experimental values from wing testing inside the indigenously developed vacuum chamber. Interesting conclusion from this study is that the CNTs do not increase the fundamental frequency, stiffness and density, but increases the strength of wings significantly. The fundamental frequency of neat PP and PP-0.1CNT is 16 Hz which is very close the flapping range(8-15 Hz) of chosen hummingbird. Hence, by using these wings, lift and aerodynamic efficiency of hummingbird inspired MAV can be increased, as the wings flap at resonance. The damping coefficient of developed wings was determined using logarithmic decrement method. In the end, some advancements were done in the design and structure of wings to mimic the hummingbird wing. Hence, PP-CNT composite wing was developed which has potential application in a hummingbird-type MAV.

ACKNOWLEDGEMENTS

We would like to acknowledge the support of the following departments from IIT Kanpur: ME (4i lab), AE (workshop and structures lab) and MSE. We would also like to thank NP-MICAV.

REFERENCES

- [1] D.L. Raney and E.C. Slominski. Mechanization and control concepts for biologically inspired micro air vehicles. *Journal of Aircraft*, 41(6):1257–1265, 2004.
- [2] P. Zdunich, D. Bilyk, M. MacMaster, D. Loewen, J. DeLaurier, R. Kornbluh, T. Low, S. Stanford, and D. Holman. Development and testing of the mentor flapping-wing micro air vehicle. *Journal of Aircraft*, 44(5):1701–1711, 2007.
- [3] R.J. Wood. Design, fabrication, and analysis of a 3dof, 3cm flapping-wing mav. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 2007.
- [4] J. Ratti and G. Vachtsevanos. A biologically-inspired micro aerial vehicle. *Journal of Intelligent & Robotic Systems*, 60(1):153–178, 2010.
- [5] T. Nakata, H. Liu, Y. Tanaka, N. Nishihashi, X. Wang, and A. Sato. Aerodynamics of a bio-inspired flexible flapping-wing micro air vehicle. *Bioinspiration & Biomimetics*, 6(4):045002, 2011.
- [6] X.L. Xie, Y.W. Mai, and X.P. Zhou. Dispersion and alignment of carbon nanotubes in polymer matrix: a review. *Materials Science and Engineering: R: Reports*, 49(4):89–112, 2005.
- [7] A. Szentes, G. Horvath, and C.S. Varga. Mechanical properties of polypropylene/multi walled carbon nanotube composites. *Hung J Ind Chem*, 38:67–70, 2010.
- [8] B. Emmanuel, A. Sohaib, C. Philippe, and C. Philippe. *Syntheses and Applications of Carbon Nanotubes and Their Composites*. InTech Europe, Rijeka, Croatia, 2013.
- [9] T.E. Chang, L.R. Jensen, A. Kisliuk, R.B. Pipes, R. Pyrz, and A.P. Sokolov. Microscopic mechanism of reinforcement in single-wall carbon nanotube/polypropylene nanocomposite. *Polymer*, 46(2):439–444, 2005.
- [10] M. Hemmati, G.H. Rahimi, A.B. Kaganj, S. Sepehri, and A.M. Rashidi. Rheological and mechanical characterization of multi-walled carbon nanotubes/polypropylene nanocomposites. *Journal of Macromolecular Science, Part B*, 47(6):1176–1187, 2008.
- [11] M.K. Seo, J.R. Lee, and S.J. Park. Crystallization kinetics and interfacial behaviors of polypropylene composites reinforced with multi-walled carbon nanotubes. *Materials Science and Engineering: A*, 404(1):79–84, 2005.
- [12] P. Wu, B.K. Stanford, E. Sllstrm, L. Ukeiley, and P.G. Ifju. Structural dynamics and aerodynamics measurements of biologically inspired flexible flapping wings. *Bioinspiration & Biomimetics*, 6(1):016009, 2011.
- [13] N.J. Sladek, M.L. Anderson, and R.G. Cobb. Aero and structural-dynamic repeatability of a novel mav wing manufacturing process. In *AIAA Atmospheric Flight Mechanics Conference*, 2011.
- [14] S. Heathcote, Z. Wang, and I. Gursul. Effect of span-wise flexibility on flapping wing propulsion. *Journal of Fluids and Structures*, 24(2):183 – 199, 2008.
- [15] C.H. Greenewalt. *The flight of birds : the significant dimensions, their departure from the requirements for dimensional similarity, and the effect on flight aerodynamics of that departure*. American Philosophical Society, Philadelphia, 1975.

An Ecological Approach to the Supervisory Control of UAV Swarms

C. Fuchs, C. Borst*, G. C. H. E. de Croon, M. M. (René) van Paassen, and M. Mulder
Section Control and Simulation, Faculty of Aerospace Engineering, Delft University of Technology,
2629 HS Delft, The Netherlands

ABSTRACT

This research employs ecological interface design to improve the human machine interface of an existing ground control station for the supervisory control of UAV swarms. As a case study, a general ground surveillance mission with four UAVs is envisioned. An analysis of the swarming work domain is performed to generate a reduced set of means-end relations. This analysis leads to a novel design that shows predicted coverage and range of the UAVs in the swarm. An evaluation study with 10 participants showed that the new interface successfully enables operators to control a swarm of four UAVs and mitigate problems during mission execution. The results of the evaluation study showed that operators had a better system understanding and that it promoted creative problem solving activities to scenarios that could not be solved by fixed procedures.

1 INTRODUCTION

The use of Unmanned Aerial Vehicles (UAVs) has grown rapidly over the past years [1, 2]. Advances in the fields of materials and computer technology provided the means to develop UAVs for a multitude of applications such as military uses [1], search and rescue operations [3] and wild life monitoring and protection [4]. While the reasons to use a single UAV are manifold, it is often advantageous to use several UAVs that are operating as a team. For example, UAVs with different capabilities - even unmanned ground vehicles - may be required. Some tasks must be executed at different locations simultaneously and in general, swarms of UAVs can observe a larger area in a shorter time [5]. Current systems and legislation still require at least one operator if not more to be in control of a single UAV. This would make swarms of UAVs highly labor intensive and considerably more expensive. Just recently, the final demonstration of FireSwarm, a project focusing on UAVs that autonomously find fires was impaired by this legal issue [6]. However, humans do not have sufficient mental resources to manually control multiple UAVs, so support in the form of automation is required [7].

*C.Borst@tudelft.nl

Most research is devoted to solve this technical issue by focusing on improving or increasing the degree of automation [5]. This includes investigating different Levels of Automation (LOAs) [8, 9] or different forms of collaboration between operators and the automation [10]. While these studies show good results, they mostly ignore the positive influence good visualizations can have [11]. As Chen et al. [5] showed, with increasing automation, operators are put into a supervisory role, which can bring its own problems: Depending on workload, operators can “*use, misuse, disuse, and abuse*” [12, p. 230] the automation. Also, imperfect automation can significantly reduce performance [13] and when the operator does not have a proper mental model, frequent automation surprises can occur [14, 15]. When such a joint system faces unexpected circumstances, it is often only because of the adaptability of the human operator that performance can still be adequate [16].

To support that adaptability, this research focuses on the human machine interface of such a UAV swarm, by employing Ecological Interface Design (EID) [17] to improve an existing Ground Control Station (GCS). EID is a constraint based approach to interface design that concentrates on revealing the work domain’s constraints [18]. Usually, GCSs display as much low-level information as possible, leaving it to the operator to connect it to the overall goal of the mission. Using EID, these connections of both the work domain’s constraints and system properties are explicitly connected to the overall goal in the form of means-end relations. That way, it is hypothesized that the operator is supported during unfamiliar and unexpected events [19]. This approach has shown promising results, not just in process control where it originates but also in a military setting [20, 21], in car driving [22] and aviation [23, 24]. The resulting interface is expected to give operators a better understanding of the system and enable them to creatively solve arising problems, without being limited to prescribed solutions.

In this research, EID is applied to a simplified search mission consisting of four UAVs and only a subset of the means-end relations identified during the analysis will be visualized in the interface. It is decided to add information about the remaining range - in the form of the current battery level - and weather conditions - in the form of a constant and uniform wind. On top of that, their propagation through the joint mission plan to the abstract function of coverage will be added. This set of means-end relations combines both internal and

external constraints so it covers a broad basis. In addition, the exact effect of these constraints on coverage is not easily recognizable by an operator. After creating the improved interface, a human-in-the-loop evaluation study is performed to gather feedback and test how well operators can control a UAV swarm when unexpected problem jeopardize mission success.

This paper is structured as follows: First, the results of a Work Domain Analysis (WDA) are presented and an existing GCS is analyzed. Second, the aforementioned set of means-end-relations are mapped to the interface and third, the evaluation study is presented.

2 WORK DOMAIN ANALYSIS

The WDA in this research is based upon and extends Amelink's [25] Abstraction Sophistication Analysis. In this analysis, a separate WDA is performed for different levels of control sophistication, ranging from low levels such as flight to high levels such as navigation. Each level is then required for and enables the next higher level.

According to Amelink, the joint operation of multiple UAVs is the highest level of sophistication but he did not perform the actual WDA for it. The WDA in this research starts with the joint operation, specifically a generic surveillance mission with four UAVs. However, instead of focusing on one level of control sophistication, multiple levels of both abstraction and control sophistication are combined. With this more liberal approach, the elements of most interest are captured.

All four UAVs are assumed to possess autonomous navigation capabilities and be able to perform individual missions comprising of different mission elements. How exactly these capabilities are achieved is of no concern for this analysis.

2.1 Abstraction Hierarchy

Performing the WDA produces the Abstraction Hierarchy (AH) [26]. This hierarchy describes the system at different levels of abstraction - ranging from the functional purpose of the entire system to the physical form of individual components [27]. Importantly, it also shows how different elements relate to each other. That is, it shows the means-end relations, or why, what, how interrelations.

The resulting reduced AH for this case study is shown in Figure 1 and is described in the following sections.

Abstract Function

The highest level of abstraction considered for this case study is the abstract function. Following the argumentation in [25], this level consists of principles that govern the coordination between all UAVs.

Coverage refers to the sensing that is required for the mission. A high coverage means that the UAVs' sensors are gathering data from a large area in a short time, which could be achieved by flying at a high altitude.

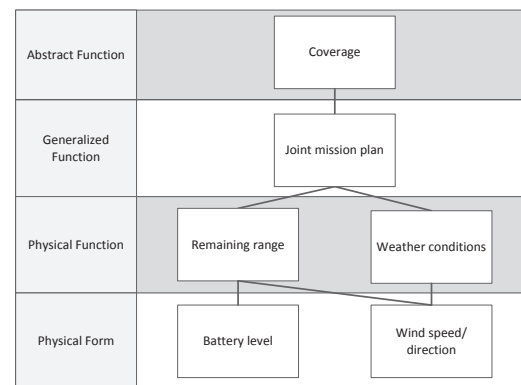


Figure 1: Reduced abstraction hierarchy

Generalized Function

The generalized functions describe how the abstract function of coverage is achieved, independent of the implementation of the system.

The means through which collaboration is achieved is the joint mission plan. This central concept of a joint mission plan is the definition of which UAV has to perform which mission at which location and at what time.

Physical Function

The physical function level is governed by constraints on the generalized function of the joint mission plan. On the one hand, it is shaped through weather conditions such as heavy precipitation. On the other hand, the remaining range of the UAVs constraints the joint mission plan. When the search area is outside the range on a certain UAV, it can obviously not be tasked with covering it.

Physical Form

Though there are several means through which the remaining range and weather conditions can be achieved, only two are considered in the interface: The battery level - or State-Of-Charge (SOC) - as well as the wind speed and direction.

2.2 Analyzing an Existing Ground Control Station

SmartUAV is a UAV research system developed at TU Delft, that is used as a test bed for advanced control of single or multiple UAVs. Development started in 2005 and has continued until now.

A possible layout for the GCS when controlling multiple UAVs, similar to other GCS interfaces like the open-source UAV project Paparazzi, is shown in Figure 2. Due to their availability to the researchers, only these two GCS interfaces are considered. The interface in Figure 2 includes four main elements: First, a mission view with a 2D-map and controls at

different levels of control sophistication ①. Second, several flight control system indicators ②. Third, low-level flight controls and a primary flight display ③. Last, a fleet overview with a small primary flight display for each UAV ④.

The following analysis is based on this configuration for multiple UAVs in a simulated environment.

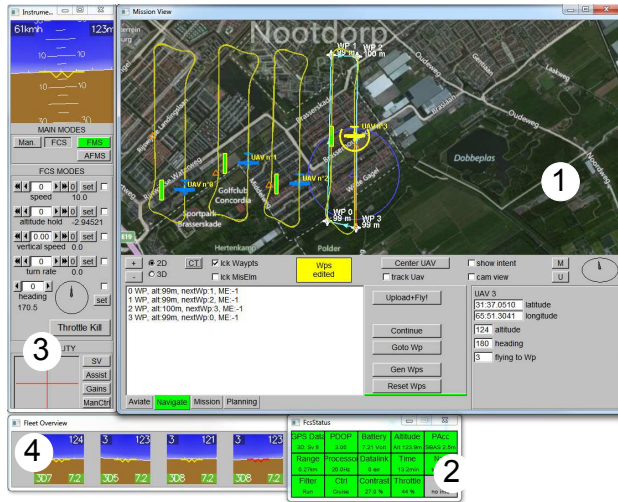


Figure 2: SmartUAV GCS for multiple UAVs, split up into the mission view ①, flight control system status ②, low level flight control ③ and fleet overview ④.

Comparing the available information in the GCS interface of SmartUAV with the set of means-end relations summarized in Figure 1, reveals that a considerable amount of these relations is not fully represented or completely missing. The results are given below, ranging from high to low levels of abstraction.

The abstract function of coverage is only partially shown. While the past track of all UAVs is visible, future waypoints are only shown for the selected UAV. Even so, this leaves the task of converting the flight path to a clear form of coverage to the operator. The same set of information is also representing the joint mission plan. By only seeing the flight plan of one UAV at a time, the operator presumably has a difficult job of integrating the information to a joint mission plan. One level lower, neither the remaining range, nor weather conditions are shown. At the lowest level, information about the selected UAV's battery level is given in terms of the voltage. Given the highly nonlinear relation between battery voltage and charge, this only gives a very crude estimate. Information about the current wind speed and direction is not included.

3 VISUALIZATIONS

Combining the WDA and the mathematical foundation of the problem, a set of visualizations is created. As there is no predefined procedure to follow, this part of the ecological approach is sometimes referred to as overcoming the creative

gap. The basis for all visualizations is a prediction of how much energy a certain flight plan will use, how much energy is left in the batteries at each waypoint and how that influences the higher level of coverage. Following are the visualizations that have been created, sorted from high levels to low levels of abstraction.

3.1 Abstract function

To visualize the abstract function of coverage, a shaded area around the flight plans of all UAVs is used, as shown in Figure 3(a). By using different shades, it is possible to show different states of coverage. Areas that will be covered are shaded lightly and areas that have been covered are shaded dark. Those areas that cannot be covered leave a "hole" in the shading, e.g., between waypoints 6 and 7 of UAV#2 in Figure 3(a).

The size of the shaded area depends on the altitude of waypoints, i.e., a larger area is shaded at a higher altitude. When the expected SOC at a waypoint is zero and the waypoint can therefore not be reached, no shading is applied. This gives the operator a clear cue that something is amiss and further fault diagnosis is required.

3.2 Generalized function

By showing the flight plan of all UAVs simultaneously like in Figure 3(a), the joint mission plan is shown. This indication is expended by applying a distinct coloring scheme to both the waypoints and the lines connecting them. Using this coloring, information about the expected SOC at every waypoint is given. This is done using three different colors according to the following rules:

- White: This waypoint can be reached and the UAV can return to base afterwards
- Yellow: This waypoint can be reached but the UAV cannot return to base once it arrives at this location
- Red: There is not enough energy in the battery to reach this waypoint

The color of a line connecting two waypoints is based upon the remaining SOC at the next waypoint. As a result, long stretches of a flight plan may be marked as unreachable, even though it is possible to reach part of them. In Figure 3(a), this can be seen between waypoints 6 and 7 of UAV#2. For illustration purposes, the exact location at which the battery will be empty is marked by "0 %". To increase safety, this conservative approach is chosen.

The means-end relation between the joint mission plan and coverage is depicted by drawing the colored waypoints on top of the coverage shading. That way, the operator can see which parts of the flight plan lead to a certain coverage, but also which waypoints fail to do so.

3.3 Physical function

Figure 4 shows how remaining range can be visualized by drawing a circle around the current position or a future waypoint. Assuming an instantaneous change of heading, this

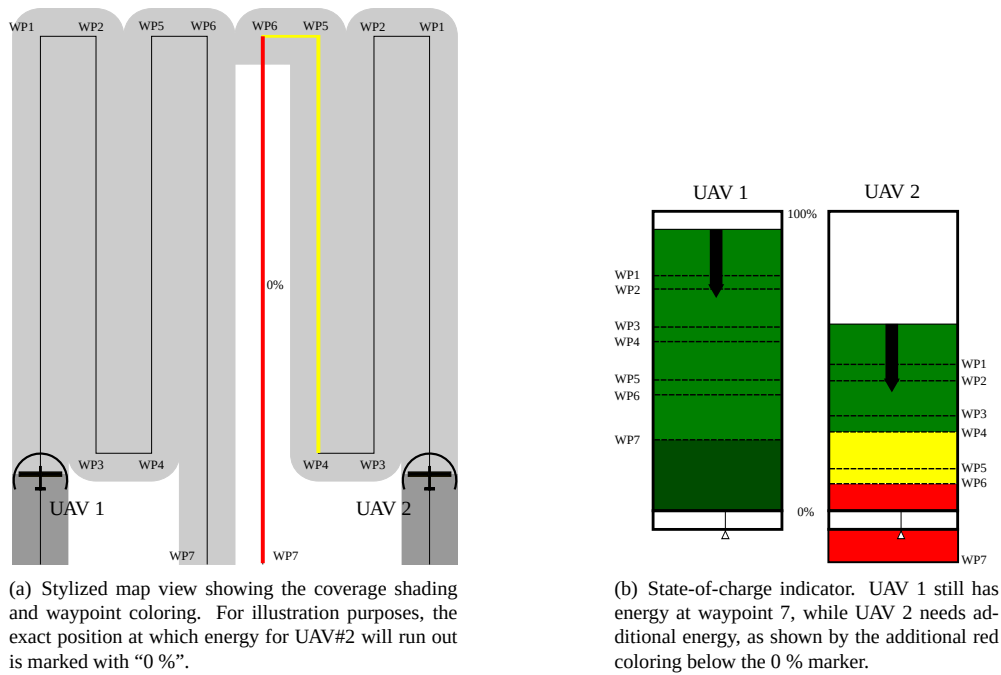


Figure 3: Side by side view of a stylized map view the state-of-charge indicator for two UAVs. Waypoint numbers (WP1 - WP7) in both depictions correspond to each other. UAV#1 is shown on the left and UAV#2 is shown on the right.

visualization is equal to a circle with radius $V_a \cdot E$ that has been shifted by $\vec{V}_w \cdot E$. Depending on whether the range from the current position or a future waypoint is shown, the airspeed and altitude used for the computation differs. At the current position, the current airspeed and altitude are used while at a waypoint, the respective values at that waypoint are taken. To reduce clutter, these circles are not shown by default and must be specifically toggled on.

A second, albeit implicit representation of the range from a future waypoint is the coloring of waypoints. A yellow waypoint shows that the remaining range is less than the distance to the base, while a red waypoint shows the remaining range is zero. This also links both the remaining range and the joint mission plan.

Usage of these range indicators is likely restricted to situations in which the operator wants to check the remaining range at a specific waypoint or compare two waypoints.

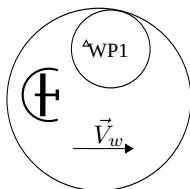


Figure 4: Remaining range from the UAV's current position and from a waypoint.

3.4 Physical form

The SOC indicator as shown in Figure 3(b) combines a set of information related to the battery level and energy usage. Most basically, the height of the colored column is an analog indication of SOC, ranging from 0 % to 100 %. Dashed lines at lower heights show the expected SOC at future waypoints, which can be lower than 0 %. In Figure 3(b), this is the case for waypoint 7 of UAV#2, so additional energy is required to reach that waypoint.

Analogue to the waypoints on the map, different colors are used to further highlight what each expected SOC means: Green represents a good status, equivalent to a white waypoint on the map, like waypoints 1 - 3 of both UAVs in Figure 3. There is, however, a distinction made between a light and dark green. The dark green, as seen below waypoint 7 of UAV#1, shows how much energy will be in the battery once the last waypoint of the flight plan is reached. Yellow shows that the waypoint can be reached but the UAV cannot return to the base afterwards, while red indicates that a certain waypoint cannot be reached at all. In order to increase safety, this coloring is done conservatively, that is the complete block towards a problematic waypoint is colored.

As an example, waypoints 1-4 of UAV#2 in Figure 3 are not problematic, as shown by the green part of the column. At waypoint 5, the UAV's SOC will have dropped so far that a safe return to the base is impossible. Therefore, the complete block between waypoints 4 and 6 is colored yellow. Simi-

larly, waypoint 7 cannot be reached at all and would require additional energy, so the column below waypoint 6 is colored red.

Lastly, the horizontal indicator at the 0 % height shows the instantaneous power consumption. The black arrow at the top of the column gives a prediction of how much energy would be used to return to base from the current position.

Each UAV that is controlled is represented by one of these SOC indicators, which are placed next to each other. That way, the operator can easily compare the SOC of all UAVs, even though they are visualized by an analog display.

Wind information is given by a simple wind sock, as depicted in Figure 5. The wind sock rotates to show the direction of wind, while the wind speed is given as a number. This enables the operator to quickly get an estimate of wind direction and a precise measurement of wind speed. For the assumed constant and uniform wind, such a display is sufficient.

Wind speed and direction considerably affect the energy usage for a given flight plan and the result is shown in other indicators. As an example, flying upwind reduces the range compared to flying downwind, which is visualized through the range circles. Therefore, while not explicitly showing the relation between wind and the remaining range, it can be deduced.

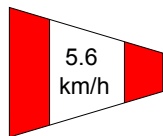


Figure 5: Wind sock

3.5 Final Interface

A screenshot of the final SmartUAV GCS interface including the aforementioned ecological improvements is shown in Figure 6. Additions to the mission view ① are the addition of the wind sock in the top left corner, coverage shading, coloring of waypoints and the ability to view all flight plans at once. Circles showing the remaining range are also added but cannot be seen in this particular screenshot. The fleet overview ② is supplemented with the SOC indicator. The flight control system status window ③ remains as it was before.

4 EVALUATION STUDY

To test if operators can indeed use the interface as intended, an evaluation study was performed. This study was carried out in the ATM-Lab of the Faculty of Aerospace Engineering, Delft University of Technology. The ATM-Lab is equipped with computers running Windows 7 on an Intel Core i7-3770 processor and an Nvidia Geforce GT 640 graphics card. In addition, a 26 inch monitor with a resolu-

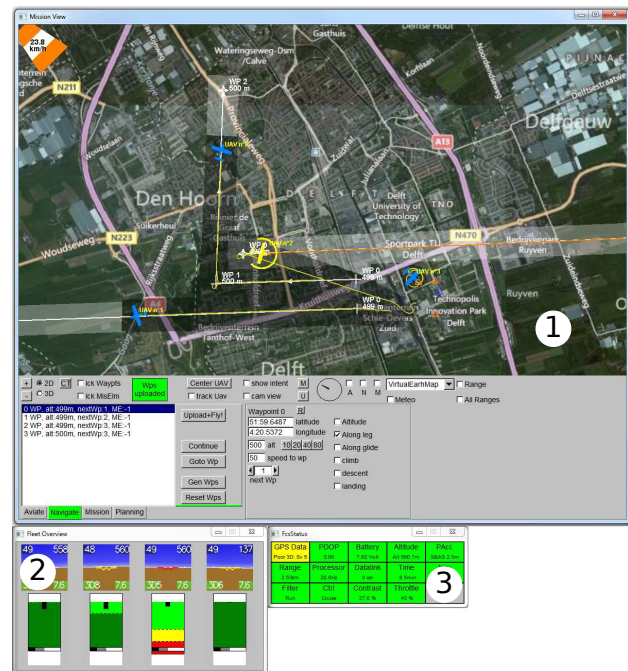


Figure 6: SmartUAV interface with ecological additions. Depicted are the mission view ①, fleet overview ② and flight control system status window ③.

tion of 2560x1600 was used to display the GCS as depicted in Figure 6.

4.1 Setup

Ten subjects - four of which have previous experience with SmartUAV - were asked to perform a mission with five different starting conditions. The objective of the mission was to survey the town of Nootdorp by loading and maintaining a predefined flight plan. This flight plan is equal to the one shown in Figure 3 but extended to four UAVs. Since pairs of UAVs are converging, this flight plan makes it easy to compensate for failures by a single UAV. Furthermore, coverage of a predefined area had to be perfect and there should be no waypoint from which a UAV cannot return to base. Possible collisions could be ignored. To reduce learning effects in the comparison of scenarios, the scenario a participant started with was alternated but the order of scenarios stayed constant. That is, two participants solved scenarios 1, 2, 3, 4, 5, two participants solved scenario 2, 3, 4, 5, 1, etc.

The five scenarios covered failures induced internally at the battery and externally at the wind condition. On top of that they covered failures at a single UAV and at multiple UAVs.

To solve problems during the mission, it was possible to change the number and position of waypoints. Participants were therefore not constrained to only use the predefined flight plan but could choose any order of waypoints. How-

ever, the altitude of waypoints was limited between 200 m and 500 m.

Scenarios 1, 3, and 5 were expected to be solved by deleting unreachable waypoints and increasing the altitude of the remaining ones. Scenario 2 contained no failure while scenario 4 was designed to be unsolvable.

4.2 Measurements

After each run, participants had to fill out a questionnaire. The first part of this questionnaire contained open questions about the participants' decision process. The second part contained a list of the improvements made to the interface that had to be rated on a scale from one (bad) to ten (good), according to their usefulness. The list of improvements that were to be rated is given in Table 1.

Table 1: Participants are asked to rate the usefulness of the following items on a scale from 1 to 10

A	Predicted coverage
B	Actual coverage
C	Coloring of waypoints
D	Coloring of lines between waypoints
E	Current state-of-charge
F	Expected state-of-charge at future waypoints
G	Energy required to return to base
H	Current power consumption
I	Current range
J	Range at future waypoints
K	Windsock

Next to the questionnaire, data was gathered by recording the screen images and voice of the participants. For that reason, participants were asked to think aloud at all times.

4.3 Results

The results of the rating feedback show some clear trends in which interface elements were considered useful or not. Figure 7 shows a bar plot of the average usefulness rating for each combination of interface element and scenario. Three of the elements appear to be the most useful, while the majority was considered not useful. Based on the usefulness rating interface elements can be put into three categories.

Very useful

- A Predicted coverage
- C Coloring of waypoints
- D Coloring of lines between waypoints

The high usefulness ratings for the predicted coverage, coloring of waypoints and coloring of lines between waypoints are in line with how the participants solved problems. Feedback to the open questions as well as the audio recording reveals that participants found and solved problems at a high level of abstraction. Specifically, the coloring was used to realize that a problem was present, while a solution was found using the coverage shading. Originally it was expected that

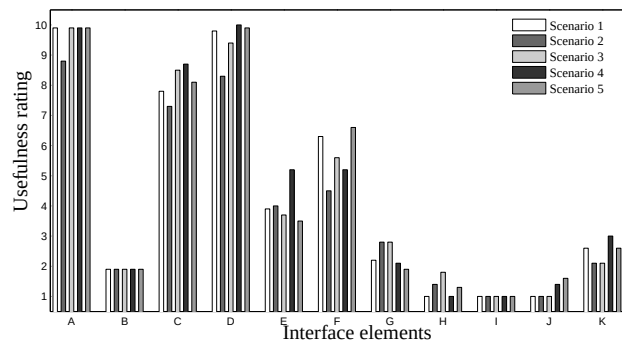


Figure 7: Bar plot of average usefulness ratings - grouped by interface element. The list of interface elements is given in Table 1.

problems are found at the highest level abstraction, i.e., the abstract function of coverage. However, adding a bright red line to the map shows to be a stronger indication than removing a light shading.

Somewhat useful

- E Current state-of-charge
- F Expected state-of-charge at future waypoints

Participants used the SOC indicator for two purposes: When the map was not centered at the search area, so that the waypoints were not visible on screen, participants used the SOC indicator to find potential problems. Mostly though, it was used to match the flight plans visible on the map with the corresponding UAVs. This reveals a considerable problem with how the joint mission plan is visualized. By showing all flight plans concurrently, without further distinction, operators were forced to use alternative means to identify the problematic UAV. Incidentally, this is the number one feedback given by participants.

Not useful

- B Actual coverage
- G Energy required to return to base
- H Current power consumption
- I Current range
- J Range at future waypoints
- K Windsock

The high number of interface elements with a low rating has different reasons. During the evaluation, problems were solved while the UAVs were en route to the search area, which implies that most of the time, the past track of the UAVs was not visible on screen. Also, participants were specifically told to ensure a perfect predicted coverage. Therefore, the low rating for the actual coverage was expected. Focusing on the prediction also caused the indication of how much energy is required to return to base to be rated low. While this indication shows more information, the yellow coloring of waypoints provides a sufficient binary feedback in the form of yes or no. As a result, the energy required to return to base is re-

dundant. However, when there was no problem as in scenario 2, this indication became more important again - an effect that carried over to the next scenario. As outlined previously, the range indicators are likely limited to very specific situations in which a direct comparison is required. During this evaluation, that was not necessary as the same information could be retrieved from the color of waypoints when they were moved around.

A surprising result is the low rating for both the current power consumption and windsock. It was expected that participants use the interface to exercise top-down reasoning to find the root cause of problems. That way, it is possible to follow the AH from failed coverage down to the battery level or wind. Instead, participants noticed and solved problems at high levels of abstraction, without looking for the cause. To give an example, during scenario 4 only one participant noticed that the wind was stronger than usual. This interface was therefore successful in not forcing a processing at a higher level of cognitive control than required.

Mission success

Out of 40 individual runs, eight were not finished successfully. Of those eight failures, four missions arguably failed due to unnecessary mistakes such as not uploading the flight plans or missing a small part of the search area. Most notably, the unsolvable scenario 4 was solved six out of ten times. Participants did so by adopting a different strategy than anticipated, which was to delete the problematic waypoints and increase the altitude of the remaining waypoints, while sticking to the general pattern of the predefined flight plan. Instead, they also changed the order of UAVs within the pattern. This shows exactly why it is important to support operators in creative problem solving.

5 DISCUSSION

The low usefulness rating of the windsock and the current power consumption was surprising, as participants were expected to use this information to find the cause of the problems. In hindsight, this behavior makes sense. With the assumption of a constant and uniform wind, the windsock does not change during the mission. Therefore, operators only needed to look at it once, which is negligible compared to other elements of the interface. In a setting with a non-constant wind, this behavior is likely to change and the usefulness of the windsock to increase. Similarly, the power usage of the UAVs only changed slightly when the UAVs were climbing and stayed constant otherwise. This indicator suffers from its small size compared to the spectrum it has to represent. So to make it more useful for operators, either the changes in power usage have to be bigger, e.g., with a different UAV, or a different visualization has to be found.

Despite the high usefulness rating of the coloring of waypoints and lines, it also received the number one comment

from all participant. By applying the same color scheme to all waypoints, operators sometimes had significant trouble finding the right UAV for the waypoints they were looking at. This makes it necessary to update how waypoints and lines are drawn. A possible solution would be to only color the lines and use different waypoint colors to indicate which waypoint corresponds to which UAV.

Future research in this field should expand the interface to include more means-end relations from the full AH, specifically the abstract function of data quality. Currently, the best solution to solve problems of coverage is to increase the altitude as much as possible. Without the restriction of a maximum altitude, that is exactly what participants would have done "because that always works". Adding data quality as a constraint should solve that issue.

6 CONCLUSION

Following an ecological approach, the human machine interface of an existing ground control station was improved to support the control of a UAV swarm. These improvements visualize how battery level (an internal constraint) as well as wind speed and direction (an external constraint) influence a higher level goal of achieving coverage in a surveillance mission. An evaluation study shows that operators can successfully use these new interface elements to control the swarm and solve problems during mission execution. The results of the evaluation study showed that operators had a better system understanding and that it promoted creative problem solving activities to scenarios that could not be solved by fixed procedures. It therefore shows that ecological interface design is a viable option to support operators of UAV swarms.

REFERENCES

- [1] S. A. Cambone, K. J. Krieg, P. Pace, and L. Wells. Unmanned aircraft systems roadmap 2005-2030. Technical report, US Department of Defense, 2005.
- [2] Parrot. Parrot establishes itself on the civil drones market, 2013.
- [3] C. Pearson. Police use drone helicopter in search, 2012.
- [4] Daily Mail Reporter. Animal welfare charity is to use DRONES to spy on people illegally hunting, 2013.
- [5] J. Y. C. Chen, M. J. Barnes, and M. Harper-Sciari. Supervisory Control of Multiple Robots: Human-Performance Issues and User-Interface Design. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(4):435–454, July 2011.
- [6] Almende B. V. Fireswarm takes technical and legal hurdles, 2013.
- [7] M. Lewis, J. Wang, and P. Scerri. Teamwork coordination for realistically complex multi robot systems.

- In *NATO Symposium on Human Factors of Uninhabited Military Vehicles as Force Multipliers*, pages 1–12, 2006.
- [8] J. C. Prinnet, A. Terhune, and N. B. Sarter. Supporting Dynamic Re-Planning In Multiple Uav Control: A Comparison of 3 Levels of Automation. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 56(1):423–427, October 2012.
- [9] M. L. Cummings and P. J. Mitchell. Automated Scheduling Decision Support for Supervisory Control of Multiple UAVs. *Journal of Aerospace Computing, Information, and Communication*, 3(6):294–308, June 2006.
- [10] M. L. Cummings, J. P. How, A. Whitten, and O. Toupet. The Impact of HumanAutomation Collaboration in Decentralized Multiple Unmanned Vehicle Control. *Proceedings of the IEEE*, 100(3):660–671, March 2012.
- [11] M. L. Cummings, A. S. Brzezinski, and J. D. Lee. Operator Performance and Intelligent Aiding in Unmanned Aerial Vehicle Scheduling. *IEEE Intelligent Systems*, 22(2):52–59, March 2007.
- [12] R. Parasuraman and V. Riley. Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 39(2):230–253, June 1997.
- [13] C. D. Wickens, S. R. Dixon, and M. S. Ambinder. 15. Workload and Automation Reliability in Unmanned Air Vehicles. In Nancy J Cooke, Heather L Pringle, Harry K Pederse, and Olena Connor, editors, *Human Factors of Remotely Operated Vehicles*, volume 7 of *Advances in Human Performance and Cognitive Engineering Research*, pages 209–222. Emerald Group Publishing Limited, 2006.
- [14] N. B. Sarter and D. D. Woods. How in the World Did We Ever Get into That Mode? Mode Error and Awareness in Supervisory Control. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):5–19, March 1995.
- [15] N. B. Sarter and D. D. Woods. Team Play with a Powerful and Independent Agent: Operational Experiences and Automation Surprises on the Airbus A-320. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 39(4):553–569, December 1997.
- [16] K. Christoffersen and D. D. Woods. 1. How to make automated systems team players. In *Automation*, volume 2 of *Advances in Human Performance and Cognitive Engineering Research*, pages 1–12. Emerald Group Publishing Limited, 2002.
- [17] J. Rasmussen and K. J. Vicente. Coping with human errors through system design: implications for ecological interface design. *International Journal of Man-Machine Studies*, 31(5):517–534, November 1989.
- [18] J. M. Flach, F. Tanabe, K. Monta, K. J. Vicente, and J. Rasmussen. An Ecological Approach to Interface Design. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 42(3):295–299, October 1998.
- [19] K. J. Vicente and J. Rasmussen. Ecological interface design: theoretical foundations. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(4):589–606, 1992.
- [20] M. P. Linegang, H. A. Stoner, M. J. Patterson, B. D. Seppelt, J. D. Hoffman, Z. B. Crittendon, and J. D. Lee. Human-Automation Collaboration in Dynamic Mission Planning: A Challenge Requiring an Ecological Approach. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(23):2482–2486, October 2006.
- [21] D. S. Hall, L. G. Shattuck, and K. B. Bennett. Evaluation of an Ecological Interface Design for Military Command and Control. *Journal of Cognitive Engineering and Decision Making*, 6(2):165–193, March 2012.
- [22] B. D. Seppelt and J. D. Lee. Making adaptive cruise control (ACC) limits visible. *International Journal of Human-Computer Studies*, 65(3):192–205, March 2007.
- [23] C. Borst, M. Mulder, and M. M. Van Paassen. Design and Simulator Evaluation of an Ecological Synthetic Vision Display. *Journal of Guidance, Control, and Dynamics*, 33(5):1577–1591, September 2010.
- [24] R. Klomp, M. Mulder, M. M. Van Paassen, and M. Roerdink. Re-design of an Inbound Planning Interface for Air Traffic Control. In *AIAA Guidance, Navigation, and Control Conference*, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, August 2011.
- [25] M. H. J. Amelink. *Ecological Automation Design, Extending Work Domain Analysis*. Phd, Delft University of Technology, 2010.
- [26] K. J. Vicente. Ecological Interface Design: Progress and Challenges. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 44(1):62–78, January 2002.
- [27] Jens Rasmussen, Annelise Mark Pejtersen, and L. P. Goodstein. *Cognitive Systems Engineering*. Wiley Series in Systems Engineering and Management. Wiley, 1994.

Modelling of a Hybrid UAV Using Test Flight Data

E.J.J. Smeur*, Q.P. Chu, G.C.H.E. de Croon, B. Remes, C. De Wagter, E. van der Horst
Delft University of Technology, Kluyverweg 1, Netherlands

ABSTRACT

The concept of an aircraft capable of both hover as well as fast forward flight (hybrid) has recently been implemented on unmanned aerial vehicles (UAV). Hybrid UAVs combine hover capability with long range and endurance. As UAVs are often required to operate without human intervention, there is a call for autonomous guidance of hybrid UAVs. Because the dynamics in the transition region from hover to forward flight are not well known, this research focusses on the development of a longitudinal model for a hybrid UAV based on test flight data. The same approach can be used for other types of airframes as well, allowing cheap and easy modelling. Sensors were logged for nine different flights and a Kalman filter was used for state estimation. The system was excited by doublet inputs on the commanded pitch and thrust. From the input-output response a piecewise linear model was estimated. This model was verified by comparing the measured doublet input response to the simulated response based on the model, yielding a very similar result.

1 INTRODUCTION

Now that Unmanned Aerial Vehicles (UAV) are adopted on a large scale, the number of applications increases as well. Many of these applications require take-off in an urbanised or otherwise obstructed environment as well as a long flight time and range. Traditional fixed wing UAVs require an open space and equipment for take-off and landing, while rotorcraft do not have as much range and endurance. A vehicle that combines the good properties of both is a *Hybrid* UAV. It can hover and transition into a fast forward flight. An example of such a UAV is the recently developed Quadshot [1].

Autonomy is a key concept for UAV aircraft, as this reduces operator workload, makes more complicated missions possible and it allows for operation beyond line of sight. Moreover, autonomy enables deployment of multiple UAVs at the same time (swarming). To accurately control the altitude and velocity of a hybrid vehicle during the transition from hover to forward flight and back, a model of the vehicle aerodynamics is necessary. There are several ways to obtain a model, a common way being wind tunnel measurements.

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

This approach has been used for hybrids as well [2, 3]. However, wind tunnel time is scarce and expensive. An alternative is identification based on test flight data, which has been done for fixed wing aircraft before [4], but not for a hybrid, which flies at a regular angle of attack in forward flight, but at very high angles of attack in the transition phase.

This paper for the first time discusses identification of a model for a hybrid aircraft based on test flights. Though this is difficult due to the poor sensor quality in a low-cost UAV and the presence of turbulence, it is a very cheap way of identifying a model. Also, it can easily be redone when the airframe is changed.

Since in the end the goal is to control altitude and velocity, a longitudinal model is needed. The inputs available are the thrust setting and the pitch command, with the assumption that the attitude is sufficiently controlled by a gain scheduled PID controller. These two inputs are sufficient for longitudinal modelling and control [5].

The test flights are performed with a Quadshot with additional sensors, which are all logged for later analysis. From this data a state estimate is calculated for each moment in time by means of an Extended Kalman Filter. From the state estimates and the recorded inputs, a piecewise linear model is constructed by least squares fitting. The obtained model is verified by analysis of the modelling error and simulations. The simulations are compared to the measured data to assess how well they represent reality.

2 EXPERIMENT SET-UP

In order to perform structured test flights, a basic autonomous flight controller was implemented in the Paparazzi open source autopilot system based on previous efforts [6]. The inner loop consist of a gain scheduled PID controller based on the transition phase. For the outer loop two cases are defined: the hover case and the forward flight case. In hover a PID controller is used for position control with thrust vectoring and the altitude is controlled using the thrust. In forward flight the position is controlled with coordinated turns and the altitude is controlled with the pitch angle. In between these flight regimes the control inputs of both controllers are interpolated.

This controller provides decent performance, but the model is expected to improve on this. The way the experiment is conducted is by flying back and forth between two waypoints. The waypoints are more than 100 m apart, so that in flying from one to the other, the Quadshot spends enough time in forward flight that the system can be excited to obtain an input response.

The Quadshot carries a 2200 mAh battery, which allows it to fly for about ten minutes. This is why the measurement procedure is optimised as much as possible. When the operator gives the launch command, the Quadshot autonomously takes off and flies to an altitude of 40 m. Upon reaching this altitude, it will proceed to the first waypoint. When it has arrived there, it waits for the command from the operator to fly back and forth to the second waypoint. Before giving this command, the operator specifies the pitch angle the Quadshot will fly at when flying forward. This ensures enough coverage of the relevant flight envelope.

The system has to be excited to make the dynamics visible. The available inputs are the thrust setting and the pitch setpoint. Doublet inputs are used on one of these inputs or both at the same time. Typically, the operator will select different doublet inputs during a flight. Finally, when the battery voltage gets low, the operator gives the command to land.

2.1 Apparatus

To reconstruct the state at each point in time after a test flight, sensors are necessary. The Quadshot is by default equipped with accelerometers, gyroscopes, magnetometers and a barometer. Additionally, a GPS receiver is added to provide information about the speed of the Quadshot. The GPS module is the Sparkfun GS407 with helical antenna, which delivers position information at 4 Hz with a typical accuracy of 4-5 m.

Next to that, the Quadshot is modified with a sensor boom. This sensor boom holds a Pitot tube (Eagletree airspeed v3) as well as an angle of attack sensor. This sensor is custom made from a wooden vane and an angle sensor with ball bearings (US Digital MA3 Miniature Absolute Magnetic Shaft Encoder). The vane is mass-balanced with a piece of lead on the tip. The boom extends 18 cm in front of the wing in order to reduce the effect of the rotors on the measurements.

The sensor boom is mounted under an angle of 33 degrees to optimize the airspeed measurement. The Quadshot is assumed to fly with angles of attack of 5 up to 55 degrees in the airspeed region of interest (transition regime). The expected mean of these angles is therefore 30 degrees. Obviously, the airspeed measurement is best if the angle of attack with respect to the sensor is zero. This is why the sensor boom is mounted under the angle that minimizes the mean.

2.2 System Excitation

The model parameters will be identified by flying at equilibrium conditions and then exciting the system through a control input. The input will be on one of the two controls, pitch or thrust, while the other is kept constant at the equilibrium value. Also, inputs will be given on both controls at the same time to determine the cross coupling. The control input is chosen to be a doublet, which is an often used input for model identification purposes [7]. A nice property of the doublet is that the deviation from the equilibrium flight path

will not be large, since the input is applied both ways. Each part of the doublet is 2 seconds. The magnitude of the input on the pitch is 10 degrees and for the thrust it is 300 on a scale from 0 to 9600. Both up-down as well as down-up doublets are used to provide a varied excitation of the system.

The pitch angle is taken as an actuator and not a state, because it is assumed to be adequately controlled by a PID controller and the pitch control is assumed to be a lot faster than the airspeed and altitude dynamics.

2.3 Axis Conventions

Before going into the various states, inputs and system matrices, it is important to define the axis conventions. Two reference frames will be used in this work, the body fixed frame and the North East Down (NED) reference frame.

The body frame has the origin in the center of mass of the Quadshot, which is assumed to be coinciding with position of the IMU. The X axis is defined perpendicular to the wing, in the direction of the pylons that are far apart. The body Z axis points parallel to the wing in the direction that the propellers blow. Finally, the system is completed by the right hand rule, according to which the Y axis should be parallel to the wing pointing to the right. The body axis system is shown in Figure 1. Because of these body axes, the pitch angle at hover is zero degrees and will go to -90 degrees for forward flight. However, the alpha is still defined as the incidence angle of the wing, so $\alpha = \theta - \gamma + 90^\circ$.

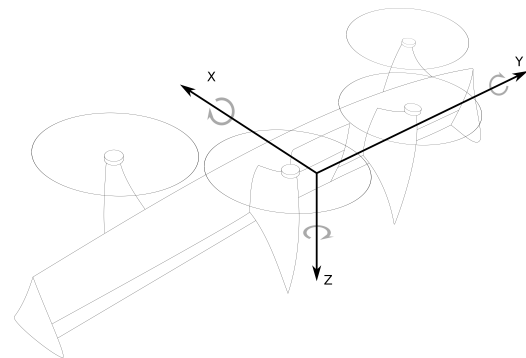


Figure 1: Body axes definition.

The origin of the NED frame is the starting point on the earth surface. The X axis points north, the Y axis points east and the Z axis points to the earth.

3 STATE ESTIMATION

The extended Kalman filter used for the state estimation is based on the extended Kalman filter published by Simon [8]. The Kalman filter consists of two stages: prediction and innovation. In the prediction stage, a prediction of the state is calculated. This is done using the state from the previous iteration $\hat{x}_{k-1|k-1}$, inputs to the system u_{k-1} , and the state equations from a system model in Eq. (1).

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}) \quad (1)$$

The calculation of the measurement residual is given by Eq. (2), where h is a non-linear mapping of the state to the measurement vector.

$$\tilde{y}_k = z_k - h(\hat{x}_{k|k-1}, u_k) \quad (2)$$

The rest of the EKF equations are quite general and will not be discussed here. The state vector used in this case is described by Eq. (3). Here q is the attitude quaternion vector and v is the velocity vector in the body frame. λ_a and λ_ω are the biases on the accelerometer and gyroscope respectively. Finally, W denotes the wind velocity vector in the NED frame.

$$x = [q \ v \ \lambda_a \ \lambda_\omega \ W]^T \quad (3)$$

The input vector u_k is given by Eq. (4). The accelerometer measurements a_a and the gyroscope measurements ω_g are used in Eq. (1) as inputs to the model. The other inputs are used as additional information in the calculation of $h(\hat{x}_{k|k-1}, u_k)$. The derivative of the GPS speed measurements gives the acceleration in the NED X and Y axes. Derivation of GPS measurements to obtain the linear accelerations is also discussed by Lopes et al. [9]. Derivation of the speed from the barometer yields the vertical acceleration along the NED Z axis. Together these accelerations form a_{NED} . Finally $q_{filtered}$ is a filtered version of the gyroscope measurement of q . For the latter four inputs no variance is taken into account, so they are filtered to remove high frequency noise. The pitch rate is filtered with a Butterworth filter with a cut-off frequency of 40 Hz to produce $q_{filtered}$. The acceleration from GPS and barometer are filtered with a cut-off frequency of four Hz.

$$u_k = [a_a \ \omega_g \ q_{filtered} \ a_{NED}]^T \quad (4)$$

The measurement vector z_k is given by Eq. (5). The measured magnetic field is denoted by B_m , the measured accelerations by a_a and the combination of GPS horizontal speed and barometer derived vertical speed by V_{NED} . V_p is the airspeed measured by the Pitot tube and α is the measured angle of attack. The GPS, airspeed sensor and angle of attack sensor have lower data rates than the IMU. To make use of every sample from the IMU, the other sensors have to be interpolated in between data samples, so that there is a data sample of every sensor at every k .

$$z_k = [B_m \ a_a \ V_{NED} \ V_p \ \alpha]^T \quad (5)$$

The mapping of the state to the measurement domain is given by Eq. (6). In this calculation the rotation matrices $T_{b0|k-1}$ and $T_{0b|k-1}$ are used, which rotate a vector from the NED frame to the body reference frame and vice versa respectively. They are constructed using the quaternion from the state $\hat{x}_{k|k-1}$. Since $T_{0b|k-1}$ is the opposite rotation,

$$T_{0b|k-1} = T_{b0|k-1}^T.$$

$$h(\hat{x}_{k|k-1}, u_k) = \begin{bmatrix} T_{b0} \cdot B_m \\ T_{b0} \cdot g + T_{b0} \cdot a_{NED} \\ T_{0b} \cdot v_b \\ \sin(33^\circ)u_a - \cos(33^\circ)w_a \\ \tan^{-1} \left(\frac{u_a - q_{filtered}d}{-w_a} \right) \end{bmatrix} \quad (6)$$

$$[u_a \ v_a \ w_a]^T = v - T_{b0} \cdot W \quad (7)$$

In Eq. (6) u_a and w_a are the body airspeeds with the wind taken into account as shown in Eq. (7). In the equation for α , u_a is corrected for the pitch rate, because the angle of attack sensor is located a distance d away from the center of gravity.

The state prediction equation is shown in Eq. (8). It shows the calculation of the derivatives and the integration of the state. The first line of the derivative calculation shows the quaternion derivative, where Ω is the rate matrix as defined in Eq. (10). Second is the derivative of the body velocity. It is composed out of the measured accelerations minus the accelerometer biases. From this measured acceleration the gravity vector rotated to the body frame is subtracted. Finally, since u , v , and w are in a moving reference frame, a correction term is present to correct for rotations. The other derivatives are zero, since their derivatives are unpredictable based on the states and inputs.

$$f(\hat{x}_{k-1|k-1}, u_{k-1}) = x_{k-1|k-1} + dt \cdot \begin{bmatrix} a - \lambda_a + T_{b0|k-1} \frac{1}{2} \Omega q + \omega \times v_b \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (8)$$

$$\omega = [p - \lambda_p \ q - \lambda_q \ r - \lambda_r]^T \quad (9)$$

$$\Omega = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \quad (10)$$

3.1 Modifications

The angle of attack measurement is found to be only valid when the norm of the airspeed is larger than 6 m/s. Therefore, the measurement covariance of the angle of attack is set to a high value when the norm of the airspeed is small. Similarly, the measurement covariance of the airspeed measurement is set to a high value when the measurement value is low, because the signal to noise ratio is very bad in that region.

Finally, the calculation of a_{NED} for Eq. (6) showed a delay of the GPS speed data. To find the right delay, the covariance of the GPS acceleration with the acceleration from the accelerometer (rotated to the NED frame) was calculated

for delays from 0 - 1 sec. For each flight, the delay that maximised the covariance was used to correct the GPS speed measurement data.

4 MODEL

The model that will be estimated is a piecewise linear aerodynamic model. Since the goal of the project is controlling airspeed and altitude, only the longitudinal dynamics are considered. Furthermore, the control of the attitude is sufficient as is, which is why the aerodynamic moment is not considered. Therefore, only the lift and drag coefficients will have to be estimated.

4.1 Model structure

According to Newton's second law of physics, the force equations in body axes are shown in Eq. (11) and Eq. (12). The aerodynamic forces are unknown functions f and g of their respective parameters. In these functions, δ_e is the input to the elevon and is assumed to be proportional to its deflection.

$$F_x = m(\dot{u} + qw - rv) = -mg \sin \theta + f(u, w, q, \delta_e) \quad (11)$$

$$F_z = m(\dot{w} + pv - qu) = mg \cos \phi \cos \theta + g(u, w, q, \delta_e, T) \quad (12)$$

The functions f and g are expected to be non-linear. To still be able to use simple but powerful linear theory, linear models will be estimated for different trim conditions. Together, these linear models will form a piecewise linear model. Doing the linearisation, the above equations can be written down as shown in Eq. (13) and Eq. (14). The acceleration in the body frame and the gravity term have been replaced by the measured acceleration. Note that $\theta, u, w, q, \delta_e$ and T will from now on be deviations from their respective equilibrium values.

$$ma_x = X_0 + X_u u + X_w w + X_q q + X_{\delta_e} \delta_e \quad (13)$$

$$ma_z = Z_0 + Z_u u + Z_w w + Z_q q + Z_{\delta_e} \delta_e + Z_T T \quad (14)$$

The model parameters will be estimated for a number of equilibrium flight conditions with different airspeeds. Between these conditions, the model parameters can be interpolated to provide a continuous model over the flight envelope.

4.2 Parameter Estimation

Eq. (13) and Eq. (14) should hold for every moment in time, so for every measurement sample. If the left hand side is written as a vector $\mathbf{y}$, the coefficients on the right hand side as a vector β , and the variables on the right hand side as matrix $\mathbf{A}$, it can be written as in Eq. (15). From this equation a least squares solution is calculated.

$$\mathbf{y} = \mathbf{A}\beta \quad (15)$$

4.3 Bins

Model coefficients are calculated for every doublet as described in the previous subsection. This results in a lot of different models for every airspeed. To reduce the number of models, the doublets are grouped into five bins of equal size ordered by the trim airspeed. For each of these bins one set of model coefficients is calculated. This way, random occurrences are averaged out and the identifiability is increased, as doublets on both inputs are taken together. For each entire bin, only one set of coefficients will be estimated, except for X_0 and Z_0 . The latter two are still estimated for each doublet, because they represent the trim condition which can differ per doublet.

If there are dependencies between the columns of $\mathbf{A}$, the matrix might be singular and the inverse does not exist. This can happen if one of the inputs is not used or certain dynamics are not excited. This problem is solved by taking different doublets together in the bins. In the calculation of the bin coefficients, 75% of the doublets are used. The remainder is saved for verification purposes.

4.4 Data Selection and Filtering

The data that will be used for the parameter estimation will be the data during the doublet only. This is because during the doublet the inputs are fixed to be either constant or the doublet input. This results in five seconds of data per doublet: one second with constant inputs, followed by a doublet with an up and a down time of both two seconds.

The accelerometer measurements that are used for the modelling contain a lot of noise from vibrations and are filtered with a Butterworth filter with a cut-off frequency of 40 Hz. The pitch rate measurement and the elevator command are filtered with a Butterworth filter with a cut-off frequency of 10 Hz. Finally, first order actuator dynamics are taken into account for the thrust. The coefficients of the first order filter were determined such that it gave a good fit with the data.

4.5 Simulation

Because in Eq. (13) and Eq. (14) the inertial acceleration is used, some rewriting is necessary to be able to do a simulation in the body reference frame. The inertial accelerations are replaced by the body accelerations along with the gravity term, as shown in Eq. (16) and Eq. (17).

$$m(\dot{u} + qw - rv) + mg \sin \theta = X_0 + X_u u + X_w w + X_q q + X_{\delta_e} \delta_e \quad (16)$$

$$m(\dot{w} + pv - qu) - mg \cos \phi \cos \theta = Z_0 + Z_u u + Z_w w + Z_q q + Z_{\delta_e} \delta_e + Z_T T \quad (17)$$

Because symmetrical flight is assumed, the terms rv , pv and ϕ are assumed to be zero. Furthermore, the gravity terms are linearised and brought to the right hand side. Then the equations are divided by m and the qw and qu terms are linearised, with the assumption that $q_0 = 0$. Then these terms

are brought to the right hand side as well. This results in Eq. (18) and Eq. (19).

$$\dot{u} = -g \cos \theta_0 \theta + x_0 + x_u u + x_w w + (x_q - w_0) q + x_{\delta_e} \delta_e \quad (18)$$

$$\dot{w} = -g \sin \theta_0 \theta + z_0 + z_u u + z_w w + (z_q + u_0) q + z_{\delta_e} \delta_e + z_T T \quad (19)$$

Where $x_* = \frac{X_*}{m}$ and $z_* = \frac{Z_*}{m}$. In state space form:

$$\begin{bmatrix} \dot{u} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} x_u & x_w \\ z_u & z_w \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} + \begin{bmatrix} -g \cos \theta_0 & x_q - w_0 & 0 & x_{\delta_e} \\ -g \sin \theta_0 & z_q - u_0 & z_T & z_{\delta_e} \end{bmatrix} \begin{bmatrix} \theta \\ q \\ T \\ \delta_e \end{bmatrix} \quad (20)$$

It can be observed from Eq. (20) that θ and q are considered as inputs. These parameters are closely related and are controlled by the attitude controller. For the performed flights, the values are determined with the Extended Kalman Filter in order to simulate the same manoeuvre as performed in the real flight. If instead this model would be used to simulate the response to an arbitrary manoeuvre, of which no θ and q data are available, the controller dynamics have to be taken into account, as well as the relation between θ and q .

5 RESULTS

The test flight experiment was conducted as described in Section 2. In total, nine flights provided useful data. This data was put through the Kalman Filter described in Section 3, of which results are shown in Section 5.1. The state estimates are used for the model parameter estimation as was described in Section 4, of which the results are shown in Section 5.2.

5.1 Kalman Filter

The Kalman filter developed in Section 3 provides stable state estimates for every performed flight. The estimated pitch angle is shown in Figure 2. Although the Kalman Filter uses quaternions, the attitude is converted to Euler angles afterwards for use in the model and for ease of perception. The figure shows the pitch setpoint angle, the estimate from Paparazzi and the estimate from the Kalman Filter. During this time segment, two doublet inputs are performed: one starts at $t = 292$ s and one at $t = 300$ s.

5.2 Model Estimation

In total, nine of these flights provided useful data. In these flights, 91 doublet inputs were given on thrust (11), pitch (37), or both (43). These inputs were given for different trim conditions. Figure 3 shows the trim pitch angle plotted against the trim airspeed. It also shows on which input a doublet was applied. From this figure a relation between pitch angle and airspeed for equilibrium flight can be observed.

The least squares parameter estimation as described in Section 4 is used on the data obtained from the Kalman filter for every doublet. This results in different model coefficients for every doublet. Plots of these coefficients are shown in

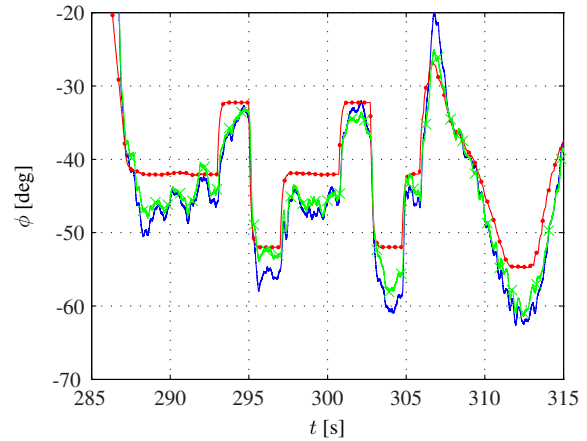


Figure 2: Pitch angle versus time. The red dotted line is the setpoint, the normal blue line is the Kalman Filter estimate and the green line with crosses is the estimate from the Paparazzi autopilot.

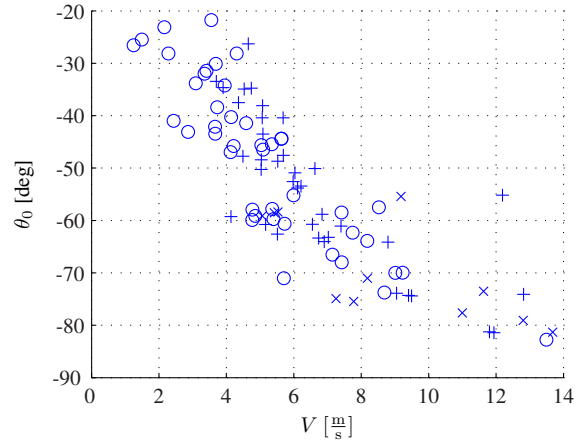


Figure 3: Doublet trim conditions. A + indicates a doublet on pitch, an x is a doublet on thrust and a o is a doublet on both.

Figure 4 and 5. Note that some of the coefficients x_T are zero. These are derived from the datasets in which only a doublet was given on pitch and the Δx_T was zero for the entire duration of the measurement.

6 ANALYSIS OF THE RESULTS

6.1 Kalman Filter

Figure 2 shows that the estimate from the Kalman Filter is close to the estimate from Paparazzi. This provides confirmation of the performance of both. Additionally, it can be seen that the estimate sometimes deviates from the setpoint more than five degrees or overshoots. This indicates that an improvement of the attitude controller might be possible. Especially, it seems that the integrator gain could be higher, since around $t = 290$ s and $t = 300$ s the same error persists for multiple seconds.

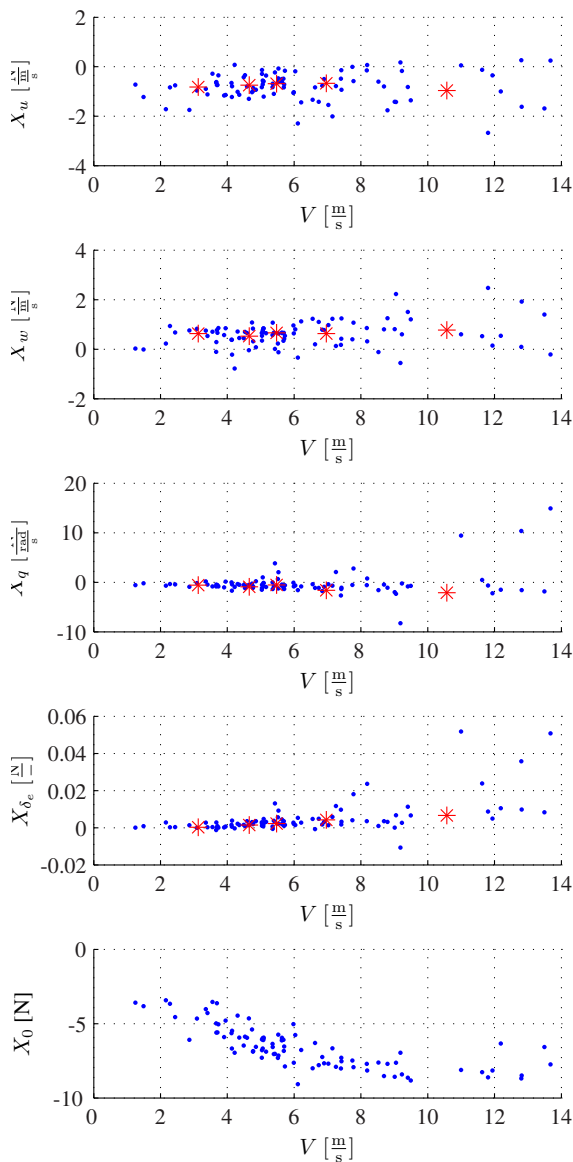


Figure 4: X axis model coefficients for every doublet. The red stars are bin coefficients.

6.2 Dynamic Pressure

Theoretically, aerodynamic phenomena scale with the dynamic pressure $\frac{1}{2}\rho V^2$. When observing Figure 4 and 5, the effect of the dynamic pressure, a quadratic trend with V , is not visible. Especially the x_u and w coefficients are quite linear. A possible explanation may lie in the poor accuracy of the estimated body velocities. Another possibility is that in a lot of the cases with high angle of attack the wing was actually stalled, yielding different results than expected.

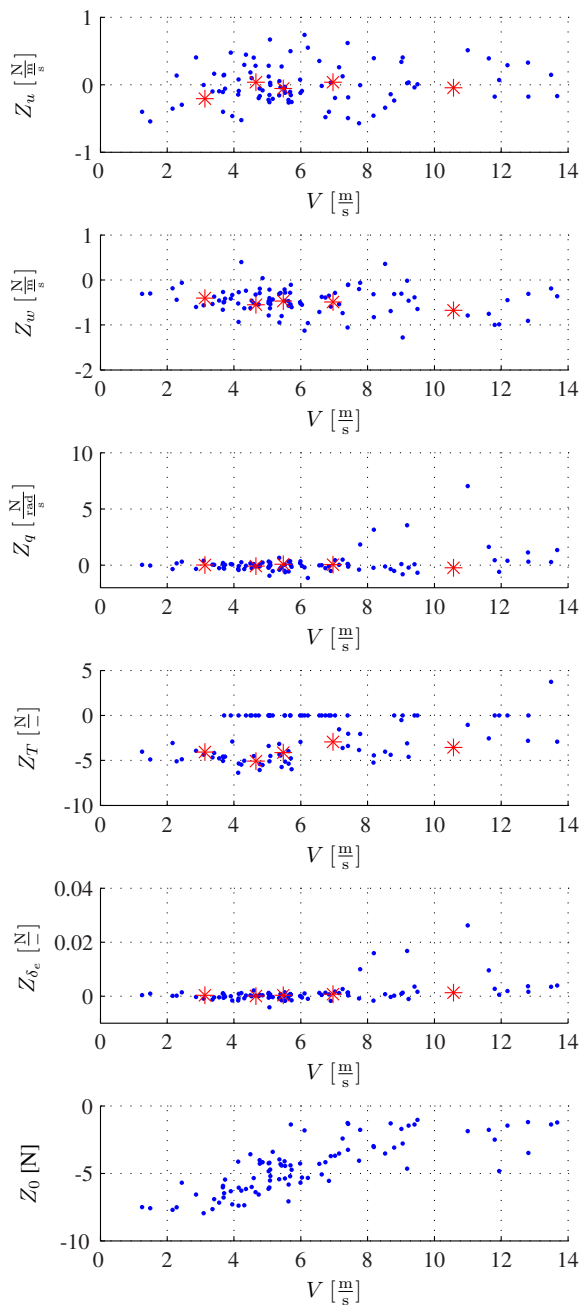


Figure 5: Z axis model coefficients for every doublet. The red stars are bin coefficients.

6.3 Trends

When looking at Figure 4 and 5, one can make some observations. Nearly for all doublets the x_u coefficient is negative. This is expected, because an increase in airspeed in the x axis will cause an increase in drag along this axis. The same holds for z_w . On the other hand, x_w is mostly positive. This can be explained as when w increases, the airspeed decreases

and less lift is produced. Depending on the pitch angle, this will generate an acceleration along the x axis.

6.4 Modelling Error

The modelling error is the mismatch between the actual acceleration and the acceleration calculated from the model, on the same dataset as used for the modelling. Figure 6 shows a plot of the measured acceleration (grey), the acceleration from the model based on just this doublet (red line with circles) and the acceleration from the model based on the second bin (blue line with crosses). From these figures it can be observed that all the main trends are modelled, but that the high frequency noise is not modelled. In this case, there is little difference between the model made specifically for this doublet and the model of the bin.

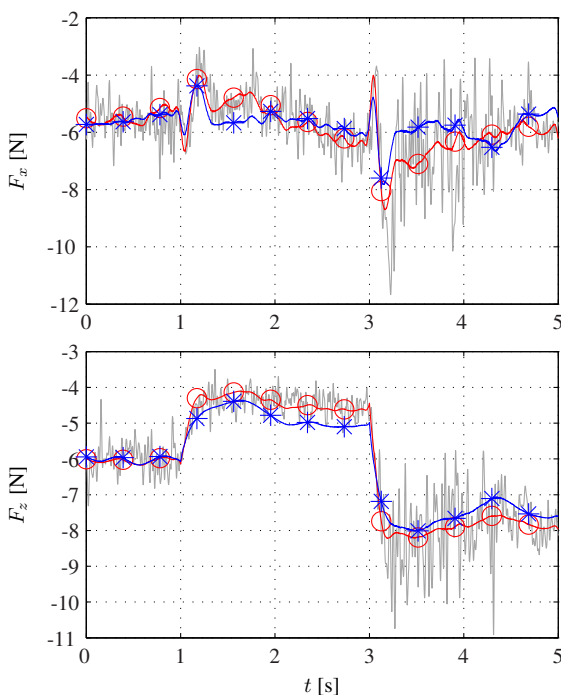


Figure 6: Model fit for an up-down pitch doublet. The grey line is the measured acceleration, the red line with circles is the best fit for this individual doublet and the blue line with crosses is the fit using the bin coefficients.

6.5 Simulation

The modelling error provides some verification, but the final test is a full simulation. This means calculation of the accelerations using the model coefficients and integrating these to get the body airspeeds. If this is done using inputs from one of the doublets, the simulated airspeeds can be compared to the recorded airspeeds.

When the linear coefficients are estimated with just one doublet, in 16% of the cases the state matrix has positive eigenvalues and an unstable model is produced. However, the

models estimated with the bins are all stable, so these are used for simulation. The way the simulations are performed is by taking the input vector in Eq. (20), $[\theta \ q \ T \ \delta_e]$, from the actual test flight data for every time sample. The model coefficients are taken from the bin the verification doublet belongs to.

Four examples of such a simulation are shown in Figure 7. These simulations are selected because of their comparable trim speeds: $2.6 \text{ m/s} < u_{\text{trim}} < 4.1 \text{ m/s}$ and $-6.9 \text{ m/s} < w_{\text{trim}} < -6.5 \text{ m/s}$. Furthermore, the same inputs are applied, a down-up pitch along with an up-down thrust doublet (the doublet goes one way at $t=1 \text{ s}$ and the other way at $t=3 \text{ s}$). The figure shows the simulated airspeed as well as the airspeed estimated by the Kalman Filter for u and w . Note again that w is positive in the direction of the 'tail' of the Quadshot. Because of visibility reasons, all simulations are depicted in blue and all measured airspeeds are depicted with dashed red lines, which means they can not be matched. It can be seen that the measurements are very noisy, but that the simulations show the same general trend as the measurements.

Clearly visible in both the simulation as well as the state estimate are the pitch changes, which cause the velocity to go from the X to the -Z axis and back. Because of the pitch down rotation w increases and u decreases. The thrust increase causes w to increase even more over time. The pitch up rotation causes velocity to shift axes from -Z to X again. After this w rapidly decreases. This can be explained because less thrust is given and a pitch up motion will increase lift and drag and therefore cause a deceleration along the -Z axis.

From a heuristic point of view, the simulation does the right thing. But the comparison with the real data is better for some simulations than for others. This might be because the accuracy of the simulation depends on the accuracy of the equilibrium condition. If the Quadshot is not in equilibrium when the doublet starts, or if the initial estimate of the trim airspeeds is incorrect, a steady state error will arise.

Because showing every verification plot is not feasible and we would like to have one number representing the accuracy of the model, a performance metric is needed. It is chosen to use the root mean squared error (RMSE) over all verification simulations. To give this number some meaning, it is compared to not having a model, resulting in simulated states $u = w = 0$ for every moment in time. In other words, the root mean square (RMS) of the u and w states from the Kalman Filter. This results in the table below.

state variable	All doublets		Figure 7	
	RMSE	RMS	RMSE	RMS
u	0.9862	1.1286	1.0557	1.1159
w	0.8432	1.1984	1.2443	1.5336

As can be seen from this table, the RMSE value is lower than the RMS value for both w and u . This means that the performance of the model is slightly better than without any model. The relatively large RMSE error might be caused by the start

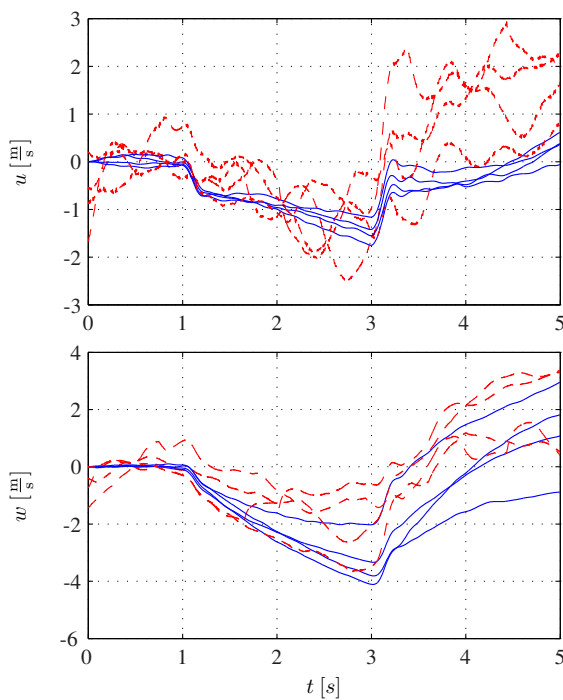


Figure 7: Four simulations of down-up pitch and up-down thrust doublets (continuous line) along with state estimates from the Kalman Filter (dashed line). $2.6 \text{ m/s} < u_{\text{trim}} < 4.1 \text{ m/s}$ and $-6.9 \text{ m/s} < w_{\text{trim}} < -6.5 \text{ m/s}$

of the doublet not being in an equilibrium condition, or an erroneous (initial) estimation of u and w .

7 CONCLUSION

Concluding, it has been shown that it is possible to identify the model of a low-cost UAV based on test flight data, avoiding the need for expensive equipment. To identify a model for the relevant flight envelope of the Quadshot, a measurement campaign was set up and carried out to obtain input response data for nine different flights. Autopilot code was written for the open source project Paparazzi to facilitate autonomous flights from take-off until landing and automation of the measurement process. A Kalman Filter was designed for the specific sensor set-up of the test Quadshot. This filter outputs a state estimate based on the sensor data. Doublet inputs were used to excite the system, resulting in 91 datasets. Five linear models were derived from the input response data, each for a different trim speed. Combined, these models provide a piecewise linear model of the Quadshot dynamics.

The model fitting was verified using analysis modelling error of the acceleration. Furthermore, it was observed that all five linear models were stable. Simulations with doublet inputs showed plausible responses, and comparison with recorded doublet responses proved the significance of the

model.

ACKNOWLEDGEMENTS

This work was supported by the Delphi Consortium.

REFERENCES

- [1] Pranay Sinha, Piotr Esden-Tempski, Christopher Forrette, Jeffrey Gibboney, and Gregory Horn. Versatile, modular, extensible vtol aerial platform with autonomous flight mode transitions. IEEE Aerospace Conference, 2012.
- [2] M. Itasse and J.M. Moschetta. Equilibrium transition study for a hybrid mav. Proceedings of the International Micro Air Vehicles conference 2011 summer edition, 2011.
- [3] Michiel J. Van Nieuwstadt and Richard M. Murray. Rapid hover-to-forward-flight transitions for a thrust-vectoring aircraft. Journal of Guidance, Control, and Dynamics, Volume 21, Issue 1, pp. 93-100, 1998.
- [4] Dongwon Jung and Panagiotis Tsiotras. Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle. AIAA Infotech at Aerospace, 2007.
- [5] Nathan B. Knoebel, Stephen R. Osborne, Deryl O. Snyder, Timothy W. McLain, Randal W. Beard, and Andrew M. Eldredge. Preliminary modeling, control, and trajectory design for miniature autonomous tailsitters. AIAA Guidance, Navigation, and Control Conference and Exhibit, 2006.
- [6] C. De Wagter, D. Dokter, G. de Croon, and B. Remes. Multi-lifting-device uav autonomous flight at any transition percentage. Proceeding of: EuroGNC 2013, 2013.
- [7] Andrei Dorobantu, Austin M. Murch, Bernie Mettler, and Gary J. Balas. Frequency domain system identification for a small, low-cost, fixed-wing uav. Proceedings of AIAA Guidance, Navigation, and Control Conference, 2011.
- [8] Dan Simon. Optimal State Estimation, page 409. John Wiley and Sons Inc., 2006.
- [9] H.D. Lopes, E. Kampen, and Q.P. Chu. Attitude determination of highly dynamic fixed-wing uavs with gps/mems-ahrs integration. AIAA Guidance, Navigation, and Control Conference, 2012.

Vortex Formation and Force Generation Mechanisms of the DelFly II in Hovering Flight

A. Tenaglia, M. Percin,* B.W. van Oudheusden, S. Deng and B. Remes
Delft University of Technology, Delft, the Netherlands

ABSTRACT

This paper addresses the unsteady aerodynamic mechanisms in the hovering flight of the DelFly II flapping-wing Micro Aerial Vehicle (MAV). Stereoscopic Particle Image Velocimetry (Stereo-PIV) were carried out around the wings at a high framing rate. Thrust-force was measured to investigate the relation between the vortex dynamics and the aerodynamic force generation. The results reveal that the Leading-Edge-Vortex (LEV), as well as the high flexibility of the wings, have a major effect on thrust generation. Comparison of three wing pairs with different aspect ratio (AR) is also reported, yielding significant differences in both vortical structures and thrust generation.

1 INTRODUCTION

The interest for Micro Air Vehicles (MAVs) is motivated by their versatility and adaptability that make them attractive for a wide range of operations. To be able to carry out these missions, MAVs require great agility and enhanced maneuverability. In this respect, flapping-wing concepts are regarded as promising systems at low Reynolds numbers, considering the abundance of flapping animal species in nature. Over the years, both biologists and engineers have investigated the mechanisms involved in insect and bird flight that may lead to flyer-based solutions for new, innovative MAV designs. In view of the inadequacy of a quasi-steady approach for the analysis of a flapping wing flight, experimental and computational research have brought new insights. There is currently consensus that the dominant aerodynamic mechanisms that govern the performance of a flapping-wing flyer are: 1) [1, 2, 3, 4] *delayed stall* and the presence of a *Leading-Edge Vortex* (LEV) which, entailing lower pressure on the wing surface, produces very high lift coefficients; 2) [5, 6, 7] *rotational circulation* and *wake capture*, mechanisms that are able to produce high peaks in force production during stroke reversal; 3) [1, 3] *clap-and-fling*, a lift-enhancing effect, which takes advantage of the wing-wing interaction at the end of the strokes.

Delft University of Technology (DUT) is represented in this research area with the DelFly, a bio-inspired small-scale

ornithopter. The first version of the DelFly was presented during EMAC '05 competition in the summer of 2005. Since that time, research and development of flapping-wing MAVs has continued at DUT.



Figure 1: DelFly II in flight.

The DelFly project follows a top-down approach, which means that by studying a large-scale model, insights can be gained to develop an improved, even smaller, version. The DelFly II (fig. 1) is the object of the current research. It has a biplane wing configuration with the wing structure composed of Mylar foil and carbon rods; the wing span is 280 mm and it weighs 17 grams. A brushless motor with a motor controller, a gear system and a crank-shaft mechanism, drive the wings in their motion. The DelFly is capable of forward and backward flight, as well as stable hovering. It is equipped with an on-board camera, which enables autonomous flight [8]. Investigation on the DelFly, has addressed different technical aspects: aerodynamical [9, 10, 11], structural [12], and the development of an analytical flight-model. Here, we concentrate on the aerodynamic mechanisms of the DelFly. A prominent feature of the aerodynamics of the DelFly is the aforementioned clap-and-fling. However because of the flexibility of the Mylar foil, the fling occurs as a peeling motion, where the wings curve along the chords and smoothly separate until the separation point reaches the trailing edge (TE). Similarly, the clap can be seen as a reverse peel. De Clercq [9] first examined the aerodynamic mechanisms involved in the hovering flight of the DelFly by means of Stereo-PIV in the vicinity of the wings in combination with simultaneous

*Email address(es): m.percin@tudelft.nl

force measurements. She stated that the peel contributes substantially to the lift generation, due to the LEV formation at the onset of the fling. In particular, the leading edge motion during the outstroke is thought to reinforce the LEV generation by increasing the velocity of the fluid moving into the opening gap. Subsequent experiments, carried out by Groen [10], confirmed that the two most important mechanisms responsible for the lift generation on the DelFly are the LEV development and the clap-and-peel effect. The instantaneous three-dimensional wake structure of the DelFly in forward flight regime was investigated by Percin and Eisma [13]. It was revealed that the wake structure is a complex interaction between tip vortices (TVs), TEVs and root vortices (RVs): a U-shaped vortical structure can be clearly seen at several instants during the flapping cycle, and its intensity increases as the reduced frequency ($k = 2fc/U_{ref}$) increases. This is consistent with the research reported by Muijres et al. [14], who revealed the existence of similar structures in the wake of bats.

Following this approach, the current research considers the generation and behaviour of the vortices involved in the DelFly motion, for a hovering flight regime. Using time-resolved Stereo-PIV and force measurements, it will be investigated how the vortical structures evolve and relate this to the force generation by the wings.

2 SET-UP

The experiments were performed in quiescent environment. Before the start of each experiment, the test room was filled with water-glycol-based particles (mean diameter of $1 \mu m$), by means of a SAFEX generator. A simplified DelFly model without tail, servo-controls, batteries and camera, was positioned on a balance structure which incorporates a 6 component force sensor (Nano 17 Titanium ATI Industrial Automation). The force measurements were recorded at a frequency of 10 kHz. Three wing pairs with the same Mylar foil thickness ($10 \mu m$) but with different aspect ratio ($AR = b^2/S$) were tested: 1) a standard wing pair ($AR=1.75$) with a wing span (R) of 280 mm and a max chord length (c) of 88 mm, 2) a high aspect ratio wing pair ($AR=2$) with a wing span of 320 mm and the same max chord length, 3) a low aspect ratio wing pair ($AR=1.5$) with a wing span of 240 mm and the same max chord length. Henceforth they will be referred to as *Std*, *HARDA*, and *LARDA*, respectively. *Std* was scaled up and down in the spanwise direction for the design of *HARDA* and *LARDA*, respectively. The stiffeners were placed with their rear tips at the corners of the wing taper and their front tips at the same relative spanwise position. Since previous experiments near the wings [9, 10] had suffered from the reflection caused by Mylar foils, the wings were sprayed with a mat black paint. This indeed improved the quality of the raw acquired images, although it did not completely eliminate the reflection problem. Two different flapping frequencies (9 and 11.2 Hz) were investigated, but in the current paper only the

11.2 Hz case will be discussed.

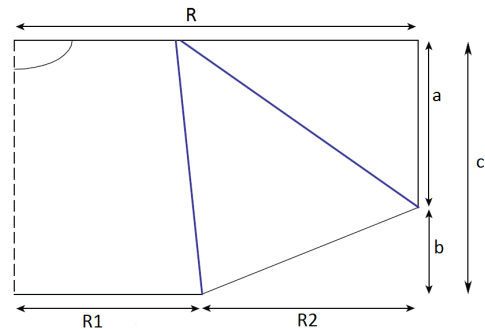


Figure 2: DelFly wing layout.

High-speed Stereo-PIV measurements were carried out at several chordwise-aligned planes around the wing (fig. 3). The first plane was placed at a distance of 40 mm outboard from the fuselage (normal to the laser sheet) and after every recording the complete balance-DelFly system was shifted by steps of 20 mm with respect to the laser sheet. Three CMOS

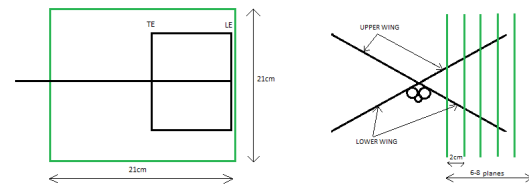


Figure 3: Schematic side view (left) and front view (right) of the measurement configuration.

cameras (High Speed Camera 6 LaVision) with a resolution of 1024×1024 pixels and equipped with a Nikon 60 mm focal objective, were used to record images. The first camera was viewing normal to the object plane, while the others were placed on the same horizontal plane, symmetrically with respect to the focal axis of the first camera, with an angle of 80 degrees between them (thus 40 degrees between each of them and the first camera). These two cameras were also equipped with Scheimpflug adapters. The choice of using three cameras for a stereoscopic approach provided the possibility of capturing both the structure within the wings and outside them during the fling phase, which otherwise would be impossible using only two cameras due to the shadows and visual obstruction by the wings. A field of view of 210×210 mm was captured with a magnification factor of approximately 0.09. Double frame image sequences of tracer particles were recorded with a frequency of 200 Hz, with a time separation between frames varying between 250 and 350 μs . The flow was illuminated with a double pulse Nd:YLF laser (Quantronix Darwin Duo) with a wavelength of 527 nm; thanks to an appropriate setting of lens and mirrors the laser

sheet was able to illuminate the entire region of interest with a thickness of 2 mm. Davis 8.2.0 software was used to control the laser and camera setting, perform image analysis, post-process and display data. The double frame images were interrogated using windows with a final size of 64 x 64 pixels and an overlap of 75%.

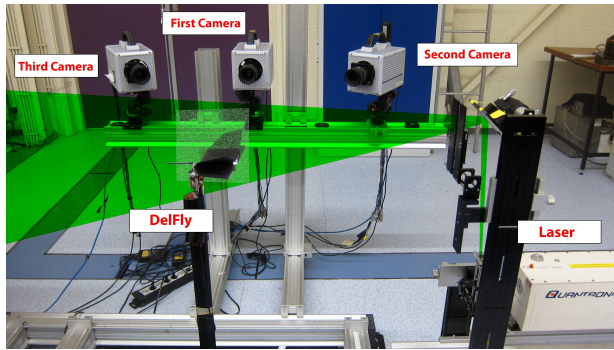


Figure 4: View of the complete set-up.

3 RESULTS

The aim of the current research is threefold: 1) to reveal the generation and subsequent shedding of the vortices from the wings of the DelFly, 2) to relate the vortex dynamics to the thrust measurements (i.e. the force component parallel to the fuselage of the DelFly) and 3) to make a comparison between the three different wing pairs.

3.1 Analysis of vortex dynamics

In this section the general vortical structure will be firstly described for the Std-wing case. By overlapping the PIV-processed vector fields with the corresponding image recordings of the second camera, it is possible to investigate the vortex behaviour while at the same time visualizing the position and shape of the wings. Different phases in a flapping cycle for a plane at 10 cm from the root are reported in the following. The vortices are labelled in order to better follow their trend throughout the images: each label reports the typology (LEV or TEV), the stroke phase (*IN* or *OUT*, for in-stroke or outstroke) and the wing to which they belong (*T* or *B*, for top or bottom). The black bands mask the regions affected by the shadow of the wings: the lack of illumination in these regions makes the evaluation of a velocity vector field unreliable. Other issues regarding the presentation of the results need to be mentioned as well. First of all, due to the presence of the force sensor, the lower region is much more affected by reflection than the upper one, and this impacts the correct representation of the velocity vectors in this region. Secondly, because of the dihedral angle, the laser sheet doesn't illuminate the wings at the same nominal spanwise position (Appendix A): this may cause that the position of vortical structures may occasionally seem to be uncorrelated

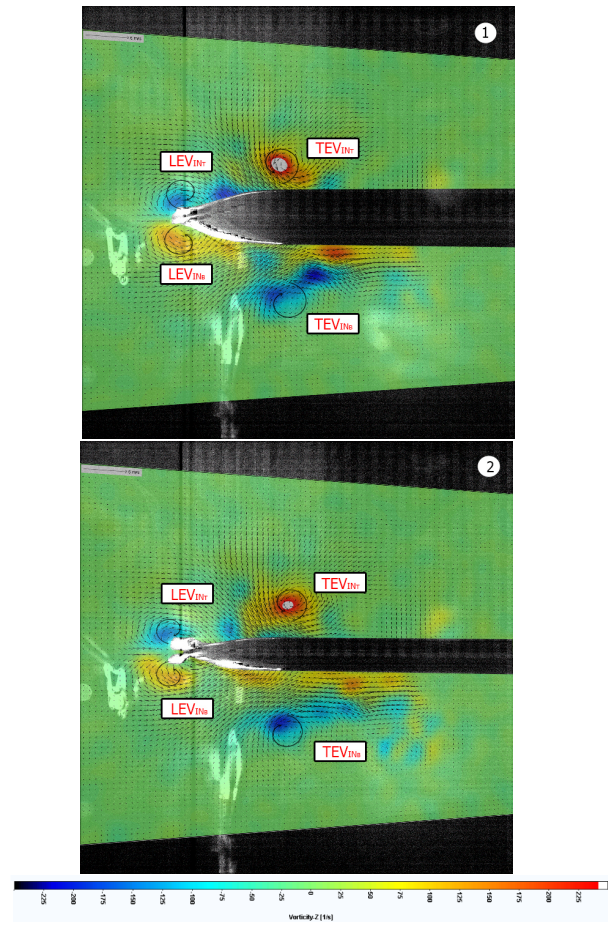


Figure 5: Two different moments of the clap mechanism. Plane located at 10 cm from the root in spanwise direction. Flapping frequency: 11.2 Hz. Time separation between images: 0.005 sec.

between the two wings. For the sake of simplicity, henceforth only the structure of the upper wing will be discussed (unless mentioned otherwise).

In Figs:5,6 four different moments of the clap-and-peel phase are presented. Starting from the top picture of Fig:5, it can be clearly seen that there are two vortices at the start of the clap, labelled as LEV_{IN_T} and TEV_{IN_T} , generated during the instroke. While the LEV_{IN_T} is still attached to the wing, the TEV_{IN_T} seems to be just shed into the wake. It then moves slowly backwards, until it is not anymore possible to track it because of the masked region (bottom picture of Fig:6). The LEV_{IN_T} , instead, moves around the leading edge and it is shed forwards: there it encounters the counter-rotating vortex from the lower wing (LEV_{IN_B}), and they presumably annihilate each other. During the initial instants of the outstroke (Fig:6), air swirls around the leading edges and rolls up into

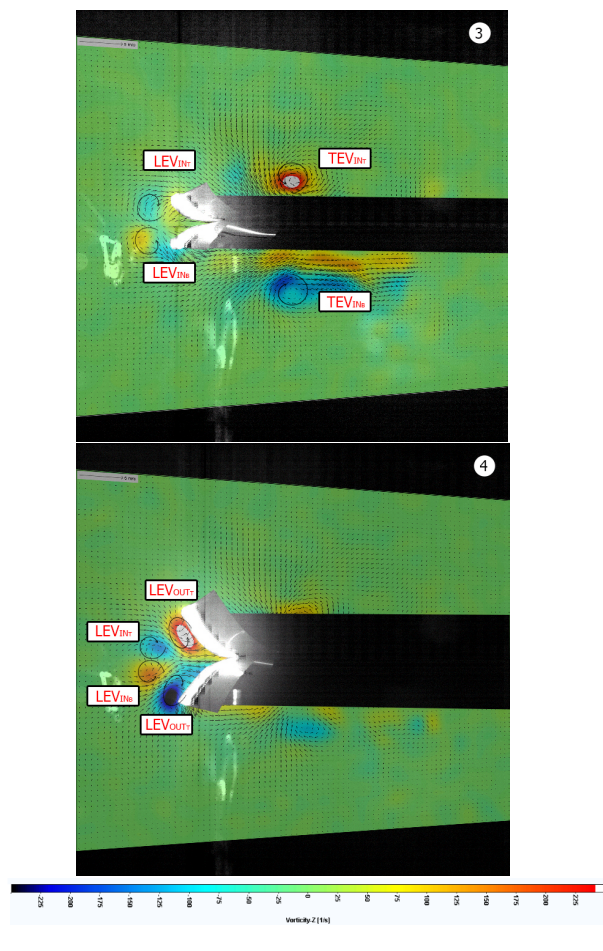


Figure 6: Two different moments of the peel mechanism. Plane located at 10 cm from the root in spanwise direction. Flapping frequency: 11.2 Hz. Time separation between images: 0.005 sec.

two intense LEVs (LEV_{OUT_T} and LEV_{OUT_B}); as a consequence, an air inrush is clearly visible in the cleft between the wings. The more the wings move away from each other (Fig:7), the more air between them acquires induced velocity and momentum because of the presence of the two vortices. The thrust experienced by the wing can be then interpreted both as a reaction to the time rate of change of this momentum and a suction force applied on the inner side of the wing itself. Approaching the end of the outstroke (Fig:7) it is again possible to see the TEV (TEV_{IN_T}) shed during the previous instroke, which still slowly moves backwards. Also (bottom picture of Fig:7), as soon as the trailing edge starts rotating around the leading edge, a new TEV (TEV_{OUT_T}) is shed into the wake (the latter will diffuse quickly and it will not be visible anymore after the start of the new instroke).

Comparing the results from different planes reveal that

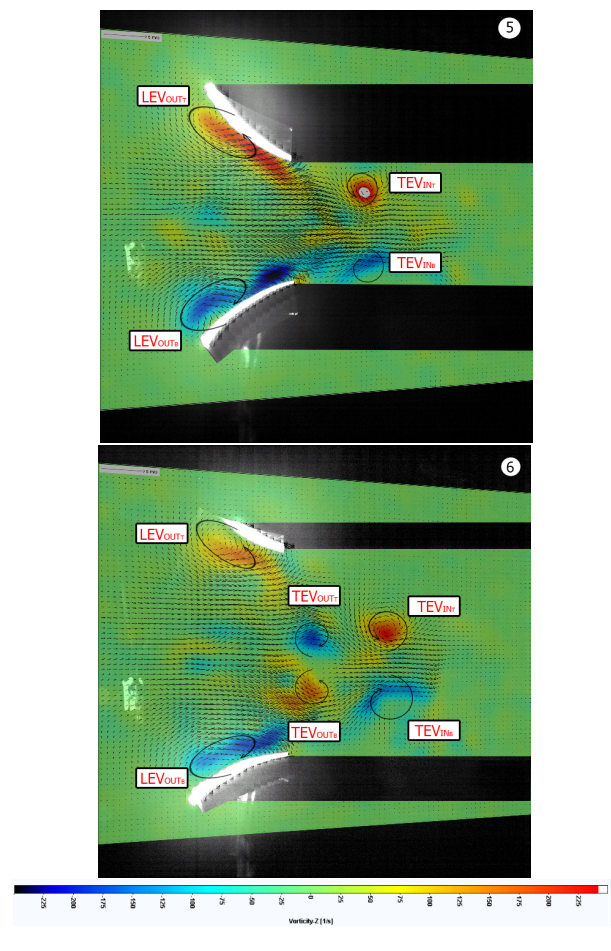


Figure 7: Two different moments in the flap cycle during outstroke. Plane located at 10 cm from the root in spanwise direction. Flapping frequency: 11.2 Hz. Time separation between images: 0.01 sec.

the general vortex behaviour briefly described so far is uniform and consistent along the wing span. However, it is worth to mention two important features that have been recognised only in the vicinity of the root, thus on the planes located at 4 and 6 cm. In Fig:8 are reported three different phases of the clap-and-peel mechanism on the plane positioned at 6 cm from the root. The TEV_{IN_T} shed during the clap soon divides into two different TEVs (labelled as TEV_{IN_T} and TEV'_{IN_T}): the TEV'_{IN_T} moves upwards and seems to disappear at the beginning of the peel, while the TEV_{IN_T} behaves like the one previously described for the plane located at 10 cm from the root. Although the cause of this separation is not yet clear, one may speculate how this phenomenon might affect the wake structure and the forces generation. However, this issue is not a topic to be debated in this paper and it will be discussed in future reports. The other feature to be men-

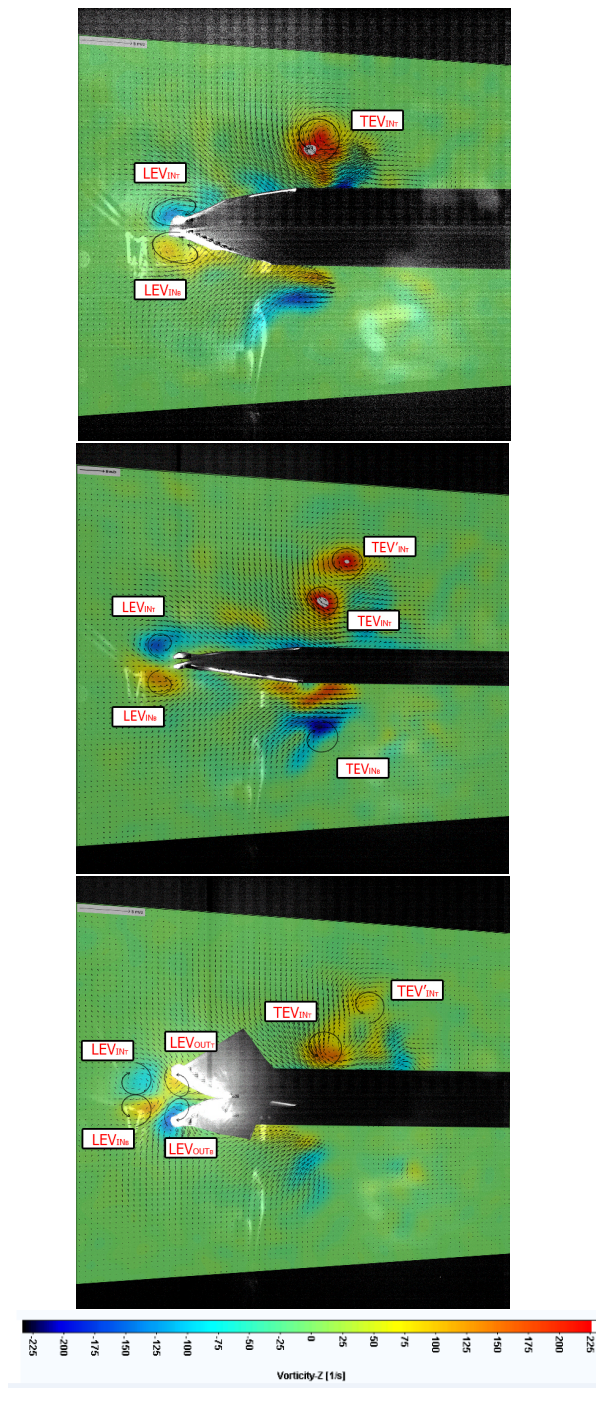


Figure 8: Three different moments of the clap-and-peel mechanism. Plane located at 6 cm from the root in spanwise direction. Flapping frequency: 11.2 Hz. Time separation between images: 0.01 sec.

tioned concerns the behaviour of the TEV_{OUT_T} . A comparison between Fig:9 and the bottom picture of Fig:7 provides a clear evidence that this vortex is no longer present in the wake of the DelFly at this spanwise position. The reason may be related to the shorter distance and a delayed separation between the trailing edges during the peel: both these effects seem to nullify the formation of the TEV_{OUT_S} .

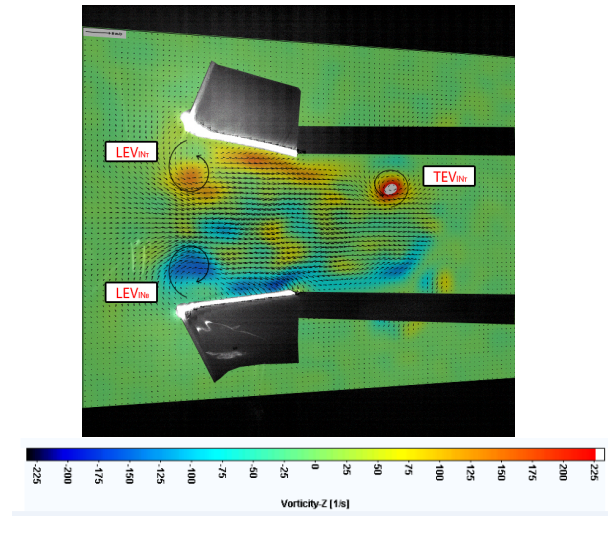


Figure 9: End of the outstroke. Plane located at 6 cm from the root in spanwise direction. Flapping frequency: 11.2 Hz.

Moving towards the tips of the wings, the laser sheet illuminates the wings only when they touch each other, thus only at the clap. For all the other phases, it is then possible to see only the vortical structures shed during the clap, while we lose sight of the vortices that remain attached to the wings. This supports the hypothesis that the TEVs (TEV_{INT} and TEV_{INB}) are shed during the clap and that they move backwards keeping the same spanwise position throughout the whole flapping cycle. Fig:10 describes behaviour at the end of the outstroke, for the plane located at 14 cm from the root. An analysis of the position of the cores of the TEV_{INT} and TEV_{INB} from 6cm to 14cm (Figs:9,7,10, respectively), reveals how these vortices are inclined to get closer to each other when approaching the wing tip. This finding suggests that there is a continuous complex structure (a TEV-tube) which is shed from the trailing edges at the clap and which is connected at the tip presumably by a tip vortex, TV.

3.2 Generation of forces

Thrust (viz. lift) is the force in the direction parallel to the fuselage of the DelFly that keeps it aloft in hover. The measured time variation of thrust over two flapping periods is shown in Fig:11 for the considered wing geometries for the flapping frequency of 11.2 Hz. The force data was filtered by means of a Chebyshev Type II low-pass filter. The cut-

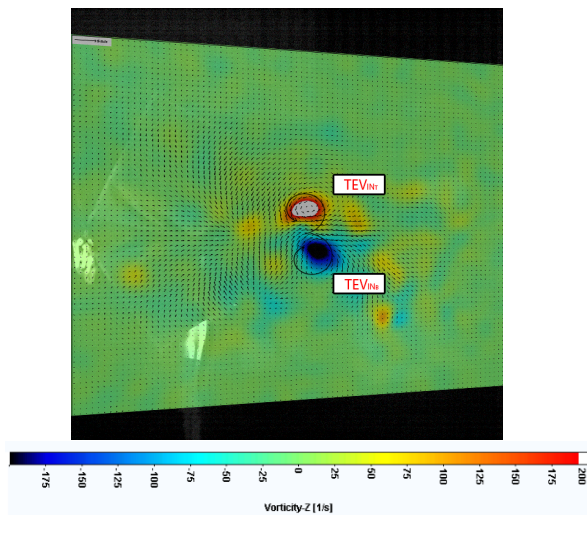


Figure 10: End of the outstroke. Plane located at 14 cm from the root in spanwise direction. Flapping frequency: 11.2 Hz.

off frequency was selected based on the flapping frequency to allow the first two harmonics of the force oscillations to be included in the resultant data as these modes were found to be related aerodynamic forces, as verified by flapping tests carried out under vacuum conditions [10]. Forces are complemented with the variation of the stroke angle (black line), flap-averaged forces and the number labels 1-6 indicate the instants when the PIV images (given in Figs: 5-7) were recorded. The start of the period defined based on the Hall sensor input and corresponds to approximately middle of the instroke.

The reciprocating nature of the flapping motion of the DelFly wings results in the formation of two peaks in the flap cycle with relatively greater thrust generated during the outstroke thanks to the enhancing effects of the clap-and-peel motion. This relative difference becomes more prominent with increasing span length of the wing (20% for LARDA, 36% for Std, and 41% for HARDA). The clap-and-peel motion starts with the approach of the wings at the end of instroke (Fig:5-1). The minimum force is generated approximately at the instant when the wings surfaces are roughly parallel to each other (Fig:5-2) and the leading edges just start the stroke reversal. At this phase, although the LEV_{INT} s are present in front of the wings, the lack of vertical wing area results in diminished force generation. Thrust rapidly increases with the start of the peel motion due to inrush of the fluid into the gap between the wings and formation of the LEVs (Fig:6-3 and 4). Force generation peaks at the middle of the outstroke (presumably when the wings have the largest deformation and thus highest effective angle of attack) and it starts decreasing with the deceleration of the wings and separation of the trailing edges (Fig:7-5 and 6).

Comparison of the mean thrust over the flapping cycle reveals that force generation increases with increasing span length, which might be attributed to the larger area of the wing surface moving with a higher velocity towards the wing tips. Thus, formation of an extended LEV in the spanwise direction with a greater circulation closer to the wing tips probably occurs and enhances the force generation. Logically, the mean power consumption also increases with the change of the span length (1.22 W for LARDA, 1.66 W for Std, and 1.84 W for HARDA).

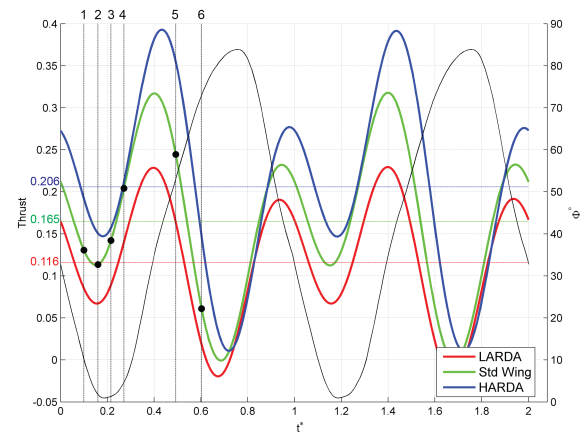


Figure 11: Time variation of thrust plotted for two periods of the flapping motion at the flapping frequency of 11.2 Hz for different wing geometries complemented with the variation of the stroke angle (black line).

3.3 Comparison between different wing pairs

In the previous section, a general description of the vortex behaviour has been presented for the Std-wing case. As shown in Fig:11, both HARDA and LARDA experience a similar thrust variation over the cycle, and, accordingly, they display the same vortical structures. Nevertheless, there are some noteworthy differences. In order to explain them, the plane at 10 cm from the root is considered, which corresponds to 83.3% of LARDA's span and 62.5% of HARDA's span, respectively.

In Fig:12 an instantaneous picture of the clap shows the velocity vector field for HARDA (top figure) and LARDA (bottom figure). It clearly shows the presence of the LEV_{INT} and the TEV_{IN_T} , whose behaviour was already described for the Std-wing case in 3.1.

The first anomalous feature is observed for HARDA-wing case: even if the laser sheet is 10 cm far from the root, the break up of the TEV_{IN_T} into two separate vortices (TEV_{IN_T} and TEV'_{IN_T}) is still present. It is likely that this behaviour is a peculiar feature of the region *relatively* close to the root, and that, as a consequence, it strongly depends on the wing

span. The latter statement is also strengthened by the fact that this TEV-split mechanism is hardly visible at 6 cm from the root for LARDA-wing case (figure not reported), where it is instead well-defined for Std-wing and HARDA cases. Moreover, it shows the presence of a vortex (labelled as LEV'_{IN_T}) in clockwise rotation, just below the TEV'_{IN_T} (top picture of Fig:12): its origin and its effect on the TEV-tube (it might be the cause of the split of TEV_{IN}) still need to be further investigated. The peel (Fig:13) confirms that the same vortical structure described before, also occurs for both HARDA and LARDA.

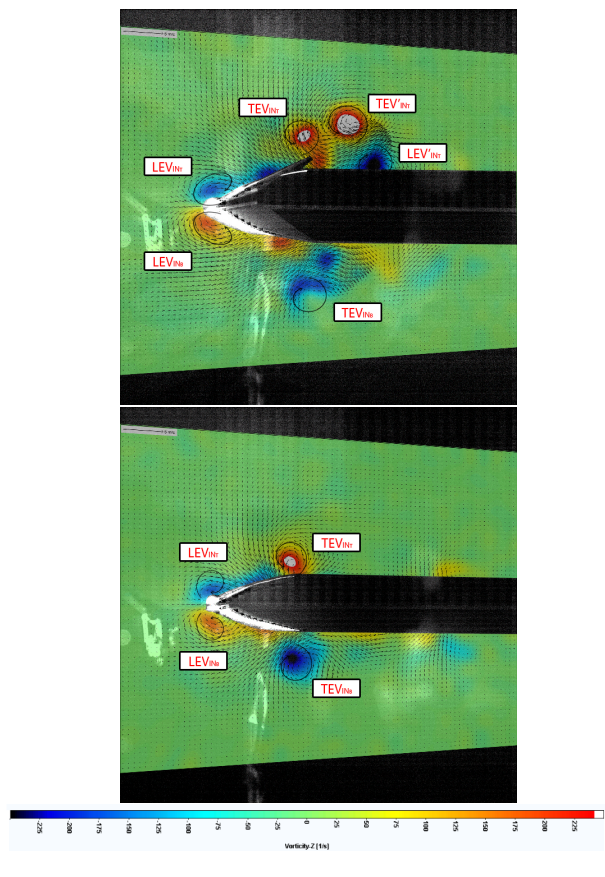


Figure 12: Comparison between HARDA (top) and LARDA (bottom), during the clap. Plane located at 10 cm from the root in spanwise direction. Flapping frequency: 11.2 Hz.

4 CONCLUSION

The Leading-Edge-Vortices (LEVs) developed during the hovering flight of the DelFly II have been shown to be strongly connected with the thrust generation. Thanks to the flexibility of the wings and to a delayed motion of the trailing-edges with respect to the leading-edges, the angle of attack of the wings reaches high values during both instroke and outstroke. This effect, combined with the existence of attached

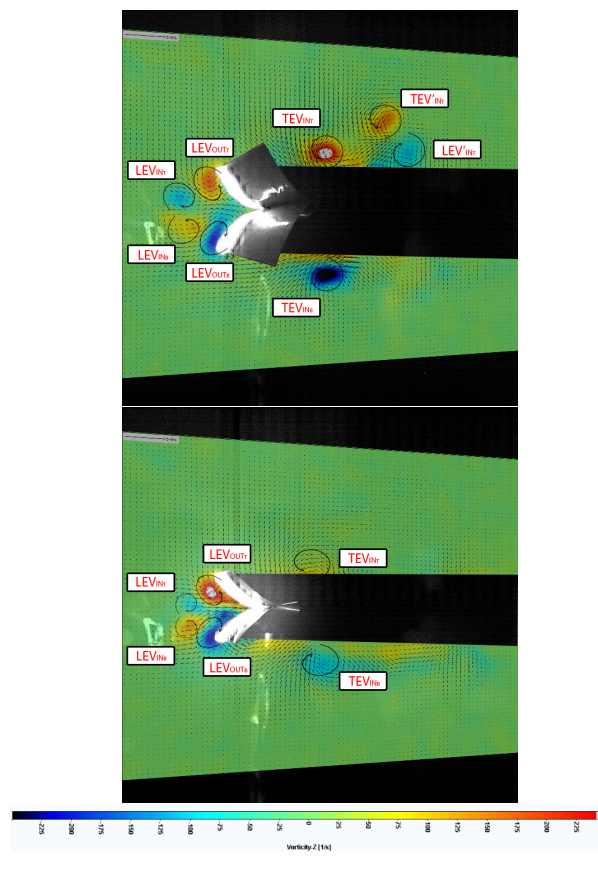


Figure 13: Comparison between HARDA (top) and LARDA (bottom), during the peel. Plane located at 10 cm from the root in spanwise direction. Flapping frequency: 11.2 Hz.

LEVs, yields strong generation of thrust-force. In contrast, because of the absence of these, the thrust-force reaches minimal values immediately after the stroke reversals. Also the Trailing-Edge-Vortices (TEVs) evolution has been hinted, but additional analysis is needed to better clarify their importance in the wake structure and force generation. Finally, a comparison between three wings with different Aspect Ratio (AR) suggested the fundamental role of the latter in forces production.

REFERENCES

- [1] C.P. Ellington. The aerodynamics of hovering insect flight. iv. aerodynamic mechanisms. *The Royal Society*, 1984.
- [2] C.P. Ellington, C. van den Berg, A.P. Willmott, and A.L.R. Thomas. Leading-edge vortices in insect flight. *Nature*, 1996.
- [3] S.P. Sane. The aerodynamics of insect flight. *The journal of Experimental Biology*, 2003.

- [4] W. Shyy and H. Liu. Flapping wings and aerodynamic lift: the role of leading-edge vortices. *AIAA Journal*, .
- [5] S.P. Sane M.H. Dickinson, F.O. Lehmann. Wing rotation and the aerodynamic basis of insect flight. *Science*, 1999.
- [6] F-O. Lehmann. The mechanism of lift enhancement in insect flight. *Naturwissenschaften*, 2004.
- [7] F-O. Lehmann, S.P. Sane, and M. Dickinson. The aerodynamic effects of wingwing interaction in flapping insect wings. *The journal of Experimental Biology*, 2005.
- [8] G.C.H.E. de Croon, K.M.E. de Clercq, R. Ruijsink, B. Remes, and C. de Wagter. Design, aerodynamics, and vision-based control of the delfly. *Internation Journal of Micro Air Vehicles*, 2009.
- [9] K.M.E. De Clercq. Flow visualization and force measurements on a hovering flapping-wing mav 'delfly ii'. Master's thesis, DUT, 2009.
- [10] M.A. Groen. Piv and force measurements on the flapping-wing mav delfly ii. Master's thesis, DUT, 2010.
- [11] J. Eisma. Flow visualization and force measurements on a flapping-wing mav delfly ii in forward flight configuration. Master's thesis, DUT, 2012.
- [12] B.Bruggeman. Improving flight performance of delfly ii in hover by improving wing design and driving mechanism. Master's thesis, DUT, 2010.
- [13] M. Percin, H.E. Eisma, J.H.S. de Baar, B.W. van Oudheusden, B. Remes, R. Ruijsink, and C. de Wagter. Wake reconstruction of flapping-wing mav delfly ii in forward flight. In *International Micro-Air-Vehicle Conference (IMAV)*, 2012.
- [14] F.T. Muijres, L.C. Johansson, R. Barfield, M. Wolf, and A. Hedenstrom G.R. Spedding. Leading edge vortex improves lift in slow-flying bats. *Science*, 2008.

APPENDIX A EFFECT OF THE DIHEDRAL ANGLE.

For all the measurements, the laser sheet is normal to the horizontal plane (see Fig:3) and its position is calculated as the distance from the root on a line (black dotted line in Fig:14) parallel to the horizontal plane. As a consequence of the flapping motion, the laser sheet does not illuminate the wing at the same position throughout the flapping cycle. Hence, in order to calculate the exact position, it must be taken into account the offset due to the stroke angle (α). Also, because of the dihedral angle (β), there will be a disparity of the laser position between the two wings. This disparity can be calculated as follows:

$$x_1 - x_2 = \frac{X}{\cos(\alpha + \beta)} - \frac{X}{\cos(\alpha - \beta)} \quad (1)$$

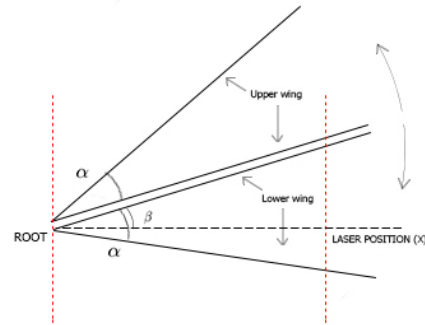


Figure 14: Effect of the dihedral angle on the effective laser sheet position on the wings.

where X is the nominal laser position, x_1 is the laser position on the upper wing, x_2 is the laser position on the lower wing and $x_1 - x_2$ is the disparity of the laser position between the two wings.

Determination of trim curves for a flapping-wing MAV

S.F. Armanini*, J.L. Verboom, G.C.H.E. de Croon and C.C. de Visser

Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS, Delft, The Netherlands

ABSTRACT

This paper presents the results of a series of flight tests conducted in order to assess the steady-state flight characteristics and basic control behaviour of the DelFly, a flapping-wing micro aerial vehicle (FWMAV). Flights were conducted in an indoor motion tracking facility and included steady-level flight at a range of different velocities and turn manoeuvres. A number of different trim points were determined and approximate trim curves constructed to describe elevator effectiveness. Aileron effectiveness was then evaluated in terms of resulting turn radii and turn rates. The results provide insight into some of the basic flight properties of the DelFly and represent a starting point for further modelling work. The flight testing process also highlighted some of the major issues to be addressed in order to obtain meaningful experimental results.

1 INTRODUCTION

Flapping-wing micro air vehicles (FWMAVs) are very small and light, and possess a number of desirable flight properties, such as high manoeuvrability, power efficiency and the capability to fly at low speeds and hover.

However, the development, modelling and autonomous control of FWMAVs is hindered by the complex nature of flapping-wing flight. Flapping-wing flyers operate at low Reynolds numbers and produce largely unsteady aerodynamics which are not yet fully understood [1, 2]. Likewise, the flapping motion of the wings implies complex, time-varying flight kinematics and can lead to significant flexibility effects [3] and inertia effects [4] that are typically negligible in fixed-wing aircraft.

Several studies have been conducted in the field of modelling and simulation of FWMAVs, e.g. with the aim of providing models for control system development or of developing virtual simulation frameworks. A small number of flight test-based studies have been conducted recently to analyse and study various aspects of flapping-wing flight including flight performance [5], kinematics and dynamics [6, 7, 8, 9], and aerodynamics [10, 11]. However, studies based on free flight data remain relatively scarce.

*Email address(es): s.f.armanini@tudelft.nl

The small size of FWMAVs greatly limits the possibility of incorporating on-board equipment such as sensors, and the range of equipment that can be used without excessively affecting flight properties. In addition to this, facilities for systematic flight testing of MAVs are not yet widely available. Flight testing is however an indispensable element in the development of any type of flight vehicle and several properties cannot be assessed by alternative means.

Within this study a series of flight tests were conducted with the DelFly, in order to determine its trim points and analyse its basic control behaviour. Combined these two components describe a fundamental component of the steady-state behaviour of the DelFly and hence represent a basis for further dynamic modelling work. Additionally, since the DelFly used for these tests has the same actuators as the autonomously flying DelFly Explorer [12], the obtained results can be used to enhance the realism of existing DelFly Explorer simulations. These simulations are used to test the DelFly Explorer's stereo vision-based autonomous flight capabilities.

This paper is structured as follows. Section 2 briefly presents the DelFly and the flight testing facilities used. Section 3 presents the flight tests conducted to find trim points and assess elevator effectiveness, and the results obtained. Section 4 outlines the tests and results for the evaluation of turn behaviour and aileron effectiveness. Section 5 concludes with a summary of the main findings and an outlook on further work.

2 PLATFORM AND EXPERIMENTAL SETUP

The subject of these tests was the DelFly (cf. Figure 1), a FWMAV developed at the Delft University of Technology. The specimen used for these tests weighs approximately 22g and has wings arranged in an 'X' configuration, with a span of approximately 290mm. The vehicle can be piloted remotely by means of a transmitter, in addition to which it is possible to hard code specific control inputs. More extensive information on the DelFly can for instance be found in [13, 14, 15].

Tests were conducted in the TU Delft *Cyberzoo*, an indoor motion tracking facility measuring 10m×10m×7m and equipped with 24 cameras (cf. Figure 2). The tracking system provides information on the position and spatial orientation of the tracked object, which is equipped with a series of either retro-reflective or LED markers. A total of 6 LED markers

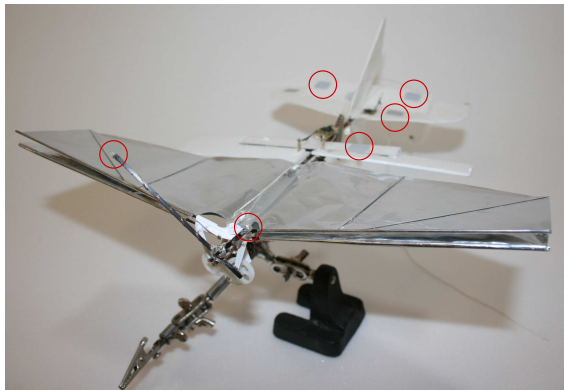


Figure 1: The DelFly with LED marker locations circled in red

were used in these tests and these were affixed at various locations on the structure of the DelFly, as shown in Figure 1.



Figure 2: The DelFly in the Cyberzoo; the motion tracking cameras are fixed to the struts shown in the image

3 TRIM CURVE DETERMINATION

The aim of this set of tests was to determine trim conditions across a portion of the DelFly's flight envelope, and to relate the input required to establish and maintain these to the resulting flight conditions. By relating control inputs to the resulting steady flight conditions, trim curves describe the steady-state behaviour of a flight vehicle across the flight envelope and are thus essential for both modelling and practical purposes.

3.1 Flight testing for trim point search

Trim conditions were defined in terms of forward flight velocity V and pitch attitude Θ . As the DelFly is intended for indoor flight and only operates within a limited altitude range, altitude can be omitted as a trim variable. The control inputs considered were the flapping frequency f_f , which is a form of thrust control, and the elevator deflection

δ_e . In practice it was found that longitudinal steady flight conditions are most effectively changed by means of elevator deflections, with the flapping frequency being used only for slight trim adjustments, and therefore flapping frequency was not used for the final trim curves at this stage.

The flight test procedure consisted in fixing the elevator at progressively changing deflection angles, covering the maximum possible range still allowing for steady flight and permitted by the testing facilities. The elevator deflection was hard-coded before each flight. The throttle, controlling the flapping frequency, was then adjusted manually by means of a transmitter to maintain a constant altitude and velocity, which also resulted in a constant pitch attitude. Whilst the aim was to explore a wide range of flight conditions, the size of the arena posed a limit to the possible range of flight velocities to investigate. At high velocities the DelFly would have covered the entire length of the arena in few seconds, leading to insufficient data, as well as entailing a high probability of crashing.

Measurements included the spatial position and attitude of the vehicle, provided by the tracking system, as well as the elevator control inputs. Given that the current test setup did not allow for actual elevator deflections to be measured, the control inputs applied by the pilot through the transmitter were measured instead.

Following the tests, pitch attitude was calculated from the measured quaternions, and velocities were obtained by numerically differentiating the measured position time series. Although ideally the tests should have been conducted at a constant altitude, evidently this is not wholly possible in real-life testing, hence to exclude the effect of any vertical velocity component, the in-plane velocity was used for analysis instead of the total velocity. Basic post-processing operations were applied to remove outliers and tracking interruptions from the data.

A total of eight flights was conducted, using the same vehicle and configuration and the same procedures. Given that the elevator deflection was hard-coded, each change in input required reprogramming and thus involved a separate flight.

3.2 Trim point search results

Table 1 summarises the results of the described tests in terms of the average values and standard deviations (reported in brackets) computed for the in-plane velocity, pitch attitude and elevator command from each of the flights. In order to obtain more accurate results, only the straight and level segments of the flights were used in the evaluation. An example of velocity measurements obtained in samples out of different flights is depicted in Figure 3.

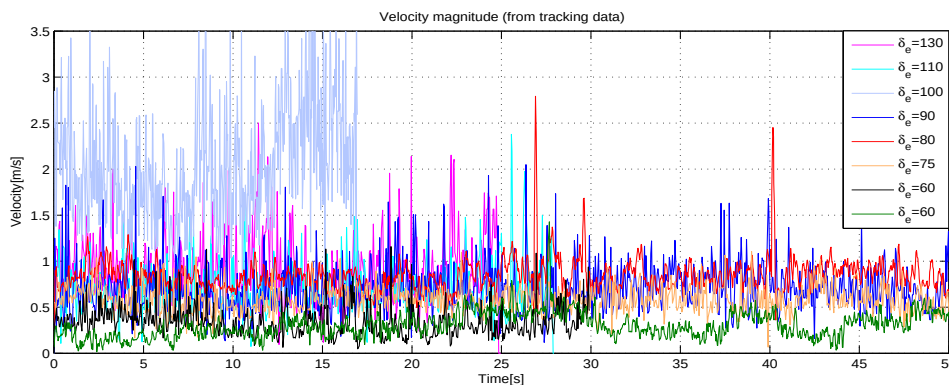


Figure 3: Velocity measurements obtained in extracts from different flight tests

The elevator commands are expressed as integer values between 0 and 250, where 0 indicates a full downward deflection. In view of the fact that elevator commands were hard-coded prior to each flight, these were assumed to be constant. In fact even a perfectly constant command does not result in an entirely constant deflection, due to vibrations and other disturbances occurring in flight, but the effects of this were assumed to be limited. Moreover, it should be emphasised that the same command value may have led to slightly different deflections during different flights, owing to external influences and changes in the hardware, e.g. as a consequence of repair work. For the same reasons, equal deflections may also have had slightly different effects during separate test runs.

Table 1: Average measurements obtained from steady level flight tests for trim point determination

Flight#	$\bar{\delta}_e [^\circ]$	$ \bar{V}_{plane} [m/s]$	$\bar{\Theta} [deg]$
1	60	0.41 (0.15)	85.12 (3.72)
2	60	0.33 (0.16)	83.86 (7.53)
3	75	0.58 (0.14)	70.54 (3.25)
4	80	0.83 (0.20)	65.95 (2.63)
5	90	0.91 (0.28)	68.47 (6.46)
6	100	2.01 (0.54)	59.57 (2.61)
7	110	0.82 (0.27)	71.45 (3.78)
8	130	1.13 (0.40)	61.09 (6.57)

It can be observed that elevator deflection is directly proportional to in-plane velocity and inversely proportional to pitch attitude. Both trends can be described relatively well by linear fit lines, as shown in Figures 4 and 5. These lines represent approximate trim curves for the DelFly, quantifying the effect of elevator input manoeuvres on its steady-state flight behaviour.

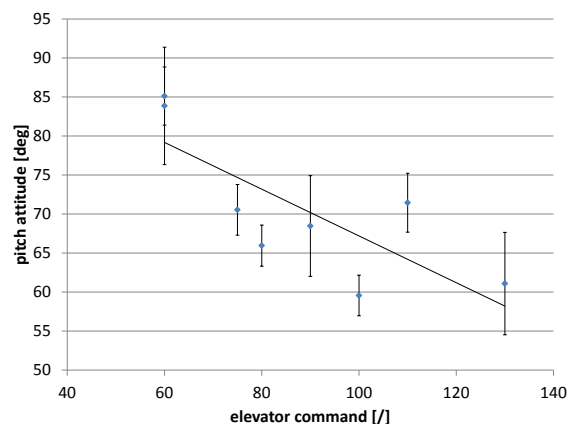


Figure 4: Trim curve: pitch attitude versus elevator command

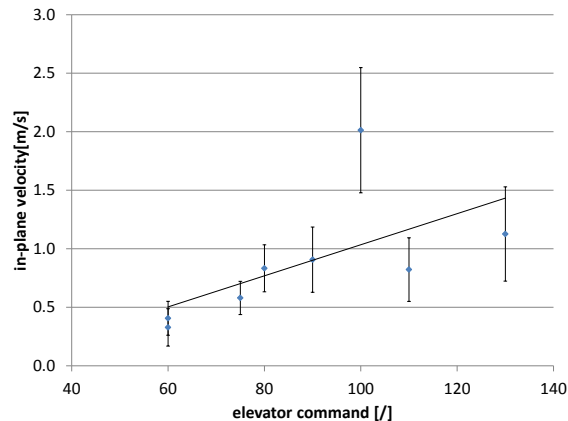


Figure 5: Trim curve: in-plane velocity versus elevator command

As can be remarked in the plots and in Table 1 (flight #6), one of the velocity measurements appears to be anomalous in the light of the remaining data. Although further testing

would be required to ascertain anything, it is assumed that the outlier was caused either by some form of external disturbance or by an unintended change in the physical state of the DelFly occurring between tests.

Besides yielding a simple input-output model for the elevator, these tests yield an overview of typical trim conditions for the DelFly, in terms of velocity and pitch attitude. Figure 6 shows that in steady level flight, velocity is inversely proportional to pitch attitude. Again, this is as expected from theory, however the quantitative results constitute some insight into DelFly's flight behaviour and provide a good starting point for further flight tests and modelling work.

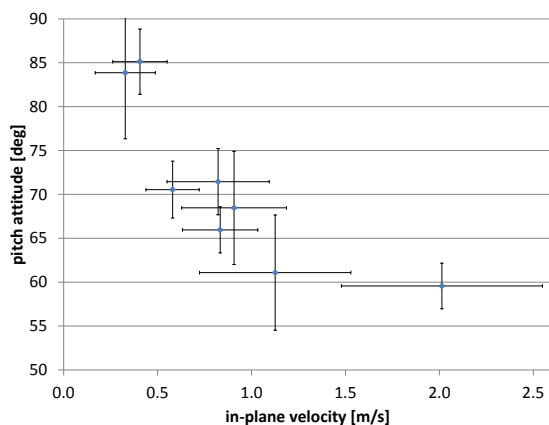


Figure 6: Experimentally found trim conditions: pitch attitude versus total velocity

4 STEADY TURNS TESTS

These tests were designed to analyse the aileron effectiveness and turn behaviour of the DelFly in terms of aileron commands and resulting turn radii and turn rates.

4.1 Flight testing and data analysis

Starting from a trimmed flight condition, progressively varying aileron inputs were applied and the ensuing turns analysed in terms of turn rate and turn radius. Aileron commands were applied manually by means of the transmitter and included inputs in both directions, in an attempt to detect any asymmetric effects.

The effectiveness of the ailerons evidently varies with the flight conditions at the time of input, thus for a complete evaluation a range of different initial flight conditions spanning the flight envelope should be used. However, in a first instance an example starting condition was chosen in order to obtain some initial insight and to evaluate the effectiveness of this type of test. Specifically, results refer to an approximate initial steady flight velocity of 0.3 m/s and pitch attitude of

80° , which is the typical flight attitude of the DelFly.

Since only a single starting condition was considered, several measurements could be conducted in the course of the same flight, by alternating between turns and straight and level flight segments. In actual fact this was not always possible due to various problems in the experimental setup, some of which will be mentioned subsequently.

Measurements included the positions and attitudes of the vehicle, and the aileron commands. Once again, only the commanded values could be measured, rather than the actual control surface deflections.

Turn rates were calculated through numerical differentiation of the yaw attitude as determined from the quaternion measurements. Turn radii were obtained by means of least squares fitting of circles through the measured flight trajectories, according to the algorithm described in [16]. For this, the turns were assumed to take place on a constant altitude plane, on which a circle could be fitted. Although some small variations in altitude did occur during the turns, in most cases these were negligible and effective fits were obtained, e.g. as shown in Figure 7. Manoeuvres involving large variations in altitude or velocity were excluded from the final evaluation.

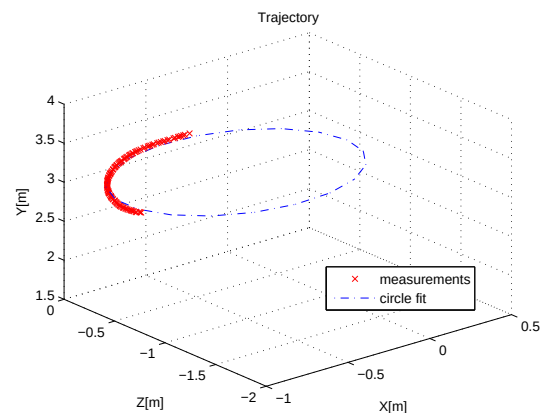


Figure 7: Example of least squares fitting of a circle to a trajectory segment during a turn

A number of problems was encountered in the process of conducting this set of tests. On the one hand, we experienced some problems with the motion tracking system, resulting in a fairly inconsistent tracking performance for these tests. The small size of the LEDs onboard the DelFly implies that the system should be set rather sensitively. Consequently, in some light conditions it is easy to obtain a higher number of either false positives or false negatives, depending on

the threshold and update frequency settings in the motion tracking system.

On the other hand, there were a number of issues with the platform itself. The DelFly has an extremely light structure, which is fragile and easily damaged. Furthermore, even slight changes to the structure, e.g. as a consequence of minor flight incidents, can affect the flight behaviour noticeably, so that the behaviour varies relatively easily and frequently to some extent. This complicates the conducting of flight tests and the evaluation of the obtained data.

In the course of the turn tests the DelFly was for instance found to have a tendency to turn in one direction in the absence of input. This in some cases caused difficulty in establishing to what extent a turn was the result of an aileron deflection, and to what extent it resulted from a previously existing tendency.

These issues must be borne in mind when considering the outcome of these tests, and make the conducting of further tests indispensable to verify the current findings.

4.2 Results

Figure 8 summarises the results of these tests, and shows aileron input magnitude versus turn radius and turn rate respectively. Aileron inputs magnitudes are given as integer values between 0 and 125, where 0 is a neutral position.

As expected, the plots suggest that turn rate is directly proportional and turn radius inversely proportional to aileron input, and both relationships appear to be somewhat linear. However there is also a considerable degree of scatter in the data. This can partly be attributed to the relatively low quality of the measurements made in these tests, which required significant post-processing operations and even then resulted in only a limited number of usable segments.

Due to the issues in the flight testing process, some of which were addressed previously, a meaningful assessment of differences between the responses to opposite sign aileron inputs was not possible. Despite these limitations, however, relatively clear trends can be recognised in the results, suggesting that further testing might provide clearer and more extensive insight and that on a high level the chosen approach is reasonably effective. The linear fits through the obtained plots allow for specific aileron inputs to be mapped to specific turn rates and turn radii, enabling an approximate prediction of the steady turn behaviour of the DelFly.

More significant results would require a larger amount of data. Furthermore, a complete description of the DelFly's turn behaviour would also require conducting the same test starting from a range of different initial trim conditions.

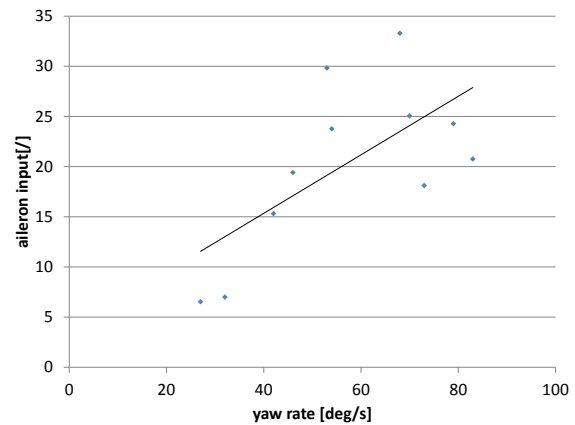


Figure 8: Aileron input versus turn rate for flight segments containing different turn manoeuvres

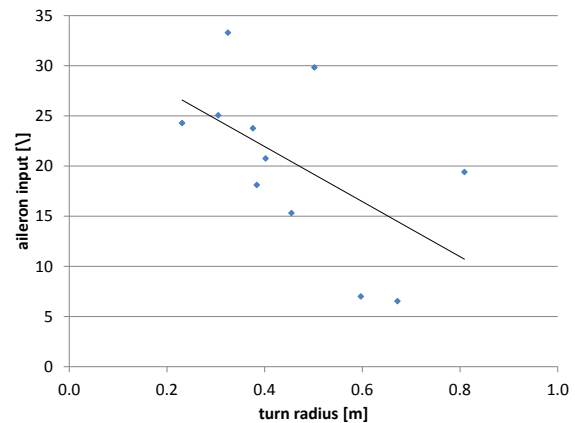


Figure 9: Aileron input versus turn radius for flight segments containing different turn manoeuvres

5 CONCLUSION

A series of simple flight tests was conducted in the TU Delft motion tracking facility *Cyberzoo*, with the objective of investigating a number of basic flight properties of the DelFly. The collected data was used to map control inputs to resulting steady flight conditions, and thereby to provide an initial overview of control effectiveness and input-output behaviour. Furthermore, a number of trim conditions were identified for the DelFly, covering a portion of its flight envelope. The results constitute a basis for more extensive modelling and can be used to lend more realism to the simulated DelFly Explorer model. Additional flight tests will be conducted to validate the current findings and add to them, e.g. by covering a wider range of flight conditions including climb and acceleration manoeuvres. More complex and comprehensive flight dynamic modelling of the DelFly is also planned.

REFERENCES

- [1] Wei Shyy, Yongsheng Lian, and Jian Tang. *Aerodynamics of low Reynolds number flyers*. Cambridge aerospace series. Cambridge University Press, 2008.
- [2] Charles P. Ellington, Coen van den Berg, Alexander P. Willmott, and Adrian L. R. Thomas. Leading-edge vortices in insect flight. *Nature*, 384(?):626–630, 1996.
- [3] Jae-Hung Han, Jin-Young Lee, and Dae-Kwan Kim. Ornithopter modeling for flight simulation. In *2008 International Conference on Control, Automation and Systems*, pages 1773–1777. IEEE, October 2008.
- [4] Christopher T. Orłowski and Anouck R. Girard. Modeling and Simulation of Nonlinear Dynamics of Flapping Wing Micro Air Vehicles. *AIAA Journal*, 49(5):969–981, May 2011.
- [5] Jong-Heon Kim, Chan-Yik Park, Seung-Moon Jun, Dae-Keun Chung, Jong-Rok Kim, Hee-Chul Hwang, Bret Stanford, Philip Beran, Gregory Parker, and Denny Mrozinski. Flight Test Measurement and Assessment of a Flapping Micro Air Vehicle. *International Journal of Aeronautical and Space Sciences*, 13(2):238–249, June 2012.
- [6] Garrett Lim, Sergey Shkarayev, Zachary Goff, Philip Beran, and Wright Patterson Afb. Studies of Flight Kinematics of Ornithopters. In *IMAV 2012*, Braunschweig, Germany, 2012.
- [7] Gunjan Maniar, Ryan Randall, Sergey Shkarayev, Zachary Goff, and Philip Beran. Kinematics of Free-flight Ornithopters. In *50th AIAA Aerospace Sciences Meeting 09-12 January 2012, Nashville, Tennessee*, pages 1–18, January 2012.
- [8] Roman Krashanitsa, Dmitro Silin, and Sergey Shkarayev. Flight Dynamics of Flapping-Wing Air Vehicle. *AIAA Atmospheric Flight Mechanics (AFM) Conference*, (August):18–21, 2008.
- [9] Jun-Seong Lee and Jae-Hung Han. Experimental study on the flight dynamics of a bioinspired ornithopter: free flight testing and wind tunnel testing. *Smart Materials and Structures*, 21(9), September 2012.
- [10] Jared Grauer, Evan Ulrich, James E. Hubbard, Darryll Pines, and J. Sean Humbert. Testing and System Identification of an Ornithopter in Longitudinal Flight. *Journal of Aircraft*, 48(2):660–667, March 2011.
- [11] Joao V Caetano, Guido C H E De Croon, Christophe De Wagter, Max Mulder, Coen C De Visser, and Bart Remes. Linear Aerodynamic Model Identification of a Flapping Wing MAV Based on Flight Test Data. pages 1–15, 2013.
- [12] Christophe de Wagter, Sjoerd Tijmons, Bart D.W. Remes, and Guido C.H.E. de Croon. Autonomous flight of a 20-gram flapping wing mav with a 4-gram onboard stereo vision system. In *IEEE Conference on Robotics and Automation (ICRA)*, 2014.
- [13] Guido C H E De de Croon, Kristien M E De Clercq, Rick Ruijsink, Bart Remes, Christophe De Wagter, and Guido C H E De Croon. Design, aerodynamics, and vision-based control of the DelFly. 1(2):71–98, 2009.
- [14] Guido C H E de Croon, Mark Groen, Christophe De Wagter, Bart Remes, Rick Ruijsink, and Bas W van Oudheusden. Design, aerodynamics and autonomy of the DelFly. *Bioinspiration & biomimetics*, 7(2):025003, June 2012.
- [15] Joao V. Caetano, Coen C. de Visser, Bart Remes, Christophe De Wagter, Erik-Jan Van Kampen, and Max Mulder. Controlled Flight Maneuvers of a Flapping Wing Micro Air Vehicle: a Step Towards the Delfly II Identification. *AIAA Atmospheric Flight Mechanics (AFM) Conference*, pages 1–15, August 2013.
- [16] I. Kasa. A circle fitting procedure and its error analysis. *Instrumentation and Measurement, IEEE Transactions on*, 1001(1):8–14, 1976.

Chapter 2

Technical Articles

<http://www.imavs.org/>

Design and development of UGS flapping wing MAVs

Muhammad Ridhwan bin Jumat, Sutthiphong Srigrarom*
University of Glasgow Singapore

ABSTRACT

This paper describes the design, build and fly University of Glasgow Singapore (UGS) flapping wing MAVs using fabrication method such as laser cutting and Rapid Prototyping. The first prototype was made from acrylic using a laser cutting machine. The material was strong however it was brittle. The wings were made up of carbon rods and kite material Ripstop. First test showed that the wings were too heavy for the mechanism to work. The second and final prototype was a smaller single gear crank design which was fabricated using a 3D printer. Initial test proved that the prototype 2 could withstand the high frequency flapping required for lift. The second test performed was to tether it on a string. At high frequency the prototype 2 was able to move in a circular motion.

1. INTRODUCTION

In recent years the development and research done on flapping wing MAV is on the rise. This is due to the technological advancement of micro components such as an electric motor. The military in particular is very interested in its development. They seek to one day provide each squad with an MAV to provide real-time intelligence during missions. Although the research is relatively new, there has been some success in the trying to mimic the flight of a bird and insect. The development of flapping wing MAV has been lagging. This is due to the complexity of the design and the unsteady aerodynamic forces of flapping wing. Only in recent years, flapping wing MAV has been gradually being picked up and researched further. The flapping wing MAV has multiple advantages compared to the other types of MAV. Flapping wing MAV is able to hover and can do so without making much noise. Its lift is more efficient compared to the other 2 MAV. However the most notable advantage is that it resembles either a bird or an insect. Be it on the battlefield or for wildlife research where blending in into the surrounding and not getting noticed is key to the success of the mission. Flapping wing MAV can be categories into: Ornithopters (bird-like flapping) and Entomopters (insect-like flapping). The ornithopter is capable of only flying forward whereas the entomopters is able to fly forward and hover as well.

2. PROTOTYPE 1

The flapping wing MAV would be based on an ornithopter design. The final design would be inspired by all the previous MAV that was researched. The design criteria is, it has to be lightweight, simple and yet strong enough to withstand the stress of the flapping motion and the crash landings during test flight. Simplicity is the key here as most of the components that would be used would be from hobby shops.

2.1 Flapping Wing Mechanism

The flapping wing mechanism function is to convert the motor's rotary motion into flapping motion. It is the most important component of the MAV thus much research was done to assess the many different designs available. Generally the mechanism design is about the same to each other with only slight modifications.

Staggered Crank Design

The staggered crank design in Figure 1 is the most basic of the flapping wing design. The connector rods are staggered in a measured distance and angle to ensure that the left and right wing are flapping symmetrically. This design is favoured by hobbyist who wants to attempt to make their own Ornithopter using household items. Modifications has to be made so that the motor can be used instead of a rubber band as its power source.

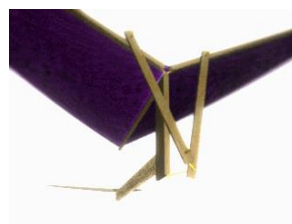


Fig. 1 (left): Staggered Crank
Fig. 2 (right): Single Gear Crank



Single Gear Crank Design

The single gear crank design in Figure 2 taken from University of California Biomimetic Millisystems Lab [13], looks simple however it is more complicated than it seems.

*E-mail address: spot.srigrarom@glasgow.ac.uk

Figure 1 shows the wings at the same level. The centre point where the connector rod and the wing hinges are connected to each other has to expand and contract as the mechanism flaps. Contracting and expanding at a very high frequency could result in component failure.

Dual Gear Crank Design

Figure 3, taken from a published paper, shows the dual gear crank design from similarly used in the Festo's SmartBird [10]. It features 2 gears that controls each wing hinges separately. There are different variation to the drivetrain design. The one shown in Figure 3, uses the pinion wheel to drive both the secondary gears. The secondary gears will rotate in the same direction with each other. The other design, has the pinion gear rotate the secondary gear and this secondary gear to another secondary gear. The secondary gears would rotate counter clockwise to each other. This design is much simpler to implement and reduce the wing symmetry misalignment.

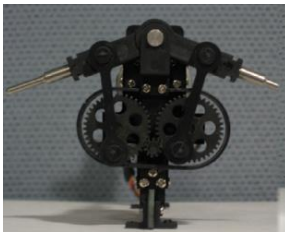


Fig. 3 (left): Dual Gear Crank
Fig. 4 (right): Transverse Shaft

Transverse Shaft

The transverse shaft design shown in Figure 4 is the other variation of flapping mechanism from Cybird 2 [6] which allows for the most symmetrical flap, however, it is the heaviest and the most complicated design. The rotating gears and the flapping wings are not in the same plane thus the connector rod has to be able to rotate. The connector rod has a ball bearing inside and this adds weight to just the component itself. The number of gears used in this design is more than any other design. The transverse shaft design is usually used for a bigger MAV design where weight could be overcome by large wings.

2.2 Tail

The tail design varies with its intended use. Some of the design uses it only for stability but in most cases they are used for control as well. For stability, the tail is tilted upwards so that the downward force of the tail would force the nose to pitch up. The angle is typically around 15 degrees or less. For control the more common designs implemented are the swinging tail and the tilting tail due to their simplicity. The swinging tail works by causing a rolling moment to when it swings to either side. The tilting tail works like a rudder, when it tilts to the right it causes the

MAV to yaw to the right. A horizontal stabilizer tail design unlike the other 2 design could provide additional control. It can act as an elevator. Providing pitch and roll control. However this design requires 2 servos to be used and a more complicated design.



Fig. 5 (left): Swinging Tail
Fig. 6 (right): Tilting Tail

2.3 Body

The body is the part where the components like the electronic speed controller, the receiver and the battery is located. The body also has to hold all the components from moving around too much. This is to prevent the shifting of the centre of gravity of the MAV. The components would each be taped separately and then hooked to the body by Velcro tape. As the design would not require much space the body design could be hollowed. Figure 7 shows the body design with holes in them. This significantly reduces the total weight of the body. The body design had to be glued to the flapping mechanism at 90 degrees angle. Small triangles were added in between them as a support structure to prevent the body and the flapping wing mechanism from snapping off.

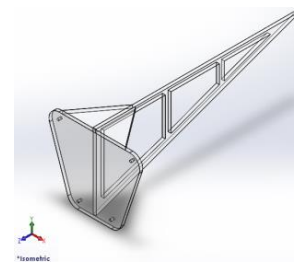


Fig. 7: Body Design

2.4 Gear and Motor Selection

Sourcing out for custom gears was a big issue. The gears that were found in the market were either too small or too big. Therefore it had to be custom design and fabricated. The gear design was dependent on the motor that is going to be used. The motor rating affects the gear ratio which then affects the flapping frequency. The formula for flapping frequency is,

$$\text{FlappingFrequency} = \frac{\text{MotorLoadSpeed}}{\text{GearRatio}} \cdot \frac{1}{60} \quad (1)$$

The motor that is used is a brushless outrunner motor. Outrunner motors have lower KV ratings meaning they have more torque but less speed. More torque is needed than speed for this project as the motors have to turn the gears to flap. The motor also needed a front mount so that it could be mounted easily to the flapping mechanism frame instead of a separate mount just for the motor. This narrows down to 2 motors as shown on the table below

Specification/Motor	Motor 1	Motor 2
Motor Rating (KV)	1200	2800
Load Speed (rpm)	5800	8350
Voltage (V)	11.1	7.4
Weight (g)	38	25

Table 1. Motor Specification Comparison

Motor 2 was chosen as it was lighter and requires lesser voltage. Voltage is linked to the number of cells that the Lithium-Polymer (Li-Po) batteries has and the rating of Electronic Speed Controller (ESC). Each cell on a battery is 3.7V so the higher the voltage the heavier the battery. It is the same for ESCs, higher ratings means bigger and heavier ESCs.

2.5 Fabrication and Material

There were 3 materials being considered: carbon fibre, balsa wood and acrylic. The first material of choice was to use carbon fibre due to it being strong and light. As it turns out, laser cutting a carbon fibre sheet would burn the material.. Balsa wood is very light and easy to cut however due to complex design of the MAV it was decided that it was not a suitable material. Acrylic was the only choice left. Acrylic is not as light and strong as carbon fibre however it can use laser cutting machine to do precision cutting.

2.6 CAD Design Dimensions

In order to find out the total dimensions and the weight that is allowed for flight, a lift equation was used. Certain assumptions that has to be made before using this equation are as follows:

1. The resulting lift would be higher in reality due to neglecting other flapping wing effects that contribute to lift when flapping.
2. The coefficient of lift is independent of the location on the wing and time.
3. Induced inflow of blade element theory is ignored.

From the assumptions made the equation for a rectangular wing shaped lift could be expanded to

$$L = \phi_0^2 \cdot \pi^2 \cdot f^2 \cdot C_L \cdot \rho \cdot c_0 \cdot l^3 \cdot \frac{1}{3} \quad (2)$$

where ϕ_0 is flapping angle, f is the flapping frequency, c_0 is the chord length and l is the wing span length. The equation is for a rectangular shaped wing however as mentioned above the equation is to be used as a rough estimate so that the dimensions and weight of the MAV could be measured. Table 2 shows the results from using the equation.

Parameters	Values	Unit
Flapping Amplitude	70	deg
Flapping Frequency	6.5	Hz
Lift Coefficient	0.8	
Air Density	1.225	kg/m ³
Chord Length	0.13	m
WingSpan	0.3	m
Lift	1.684	N

Table 2. Lift Generated

CAD Design

Using the dimensions above and the design criteria, a CAD design using SolidWorks was modelled. It would incorporate a dual gear crank and a horizontal stabilizer tail design. The dual gear crank was the simplest design with not much wing symmetry misalignment. The horizontal stabilizer tail design was chosen as it could provide both pitch and roll control. The initial design showed in Figure 8 featured a articulated wing. This design was not used as there were too many moving parts in the design and may complicate things. Therefore the chosen design is the one shown in Figure 9.

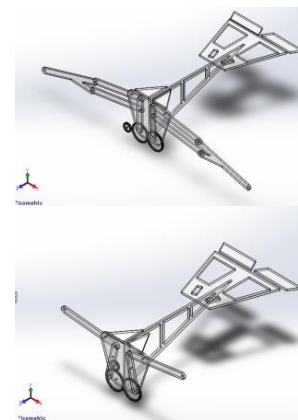


Fig. 8 (left): Articulated Flapping Wing
Fig. 9 (right): Single Flapping Wing

The total weight of the MAV was measured using one of the features in SolidWorks. Now the total weight of the MAV plus the components could be compared to the lift equation result. Table 3 shows the sum of all the component weights. The 2 measurements shows that the weight of the MAV is

below the total lift generated. An image of the assembled MAV is shown in Figure 10.

Components	Values	Unit
Brushless Outrunner Motor	25	g
Radio Receiver	11.5	g
Servos	9	g
Li-Po Battery	15	g
Electronic Speed Controller	10	g
MAV Design	80.24	g
Total Weight	150.74	g

Table 3. Total Weight of MAV



Fig. 10: Assembled Flapping Wing MAV

3. PROTOTYPE 2

Learning from prototype 1, some design considerations were made:

- the flapping mechanism needs to be more simplified
- the number of moving parts need to be reduced
- the overall design has to be much smaller to reduce weight
- changing the tail design to either a tilting or swinging tail would reduce the number of servos used which would reduce weight

3.1 Flapping Wing Mechanism

The flapping wing mechanism for prototype 1 had too many moving parts and was not simplified enough. A simpler design was needed and thus another look at the single gear crank was taken. The design idea was to shift its fixed pivot point from being at the centre of the wing to it being at the end of the 2 wing joints. Figure 11 shows this design. The changes made to its pivot point made the flapping mechanism worked properly. A simulation test was done using the software and it showed that it could hold at high frequency flapping and the flapping movement is synchronized.

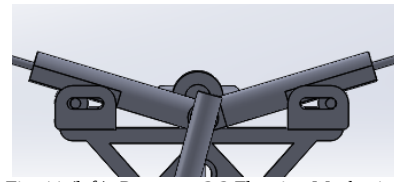


Fig. 11 (left): Prototype 2.2 Flapping Mechanism

3.2 Tail

The previous prototype was using an elevon tail design which could provide pitch and roll control however it requires 2 servos to be used. For weight reduction and simplicity sake, a simple tilting tail would be used instead. The tail frame would be made up of carbon rods which would be fixed to the tail piece and covered with Ripstop. The tail piece has a ball bearing inside it so that the tail could tilt easily. Figure 12 shows the tilting tail design.



Fig. 12 (right): Prototype 2.2 Tail

3.3 Body

In previous design the body was made out of acrylic and had to be solvent weld together. The design was simpler to implement however it was bearing a lot of weight. In order to reduce more weight, carbon rods would be connected to the front piece and the tail piece to form a rigid triangle frame. The frame would then be covered with Ripstop and Velcro tape to secure the components to the platform. Figure 13 shows the CAD design of the body.

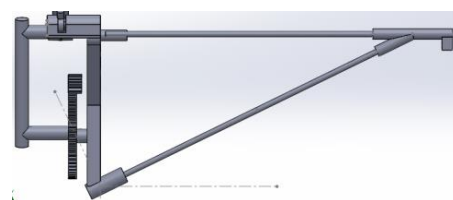


Fig. 13: Prototype 2.2 Body

3.4 Gear and Motor Selection

The new flapping mechanism uses only 2 gears. This allows more fine tuning to the gears which allowed a gear ration of 5.5:1. This was acceptable as the newer motor has a slower load speed but higher torque. The new gear is specially hollowed at the centre for a ball bearing to be inserted so that it can spin freely around the connecting part of the front piece.

Specification/Motor	Motor 1	Motor 2
Motor Rating (KV)	1700	2800
Load Speed (rpm)	7800	8350
Voltage (V)	7	7.4
Weight (g)	20	25

Table 4. Motor Specification Comparison

3.5 Fabrication and Material

In the previous design, a laser cutting machine was used. For the new prototype, a Rapid Prototyping Machine or also known as a 3D printer would be used. A 3D printer allows for more freedom of design. An extruded part could be combined during the design process easily, compared to assembling the parts after it has been fabricated. The chosen material was PLA as the design such as the gears needed the material to be strong and durable.

3.6 CAD Design Dimensions



Fig. 14 (right): CAD design Prototype 2.2

Figure 14 shows the completed CAD design of prototype 2. Once again to find out if the dimensions of the prototype could produce enough lift, the lift equation in chapter 3.6 was used and compared to the total weight of the prototype. Table 5 below shows that the lift is more than the weight thus the prototype fabrication can proceed.

Table 5. Lift & Weight Comparison

3.7 Flight test

Figure 15 shows the assembled prototype. Similarly a dry run test was done for the flapping mechanism. Everything was working normally. Next it was the tethered flight. The MAV was also able to move in a circular motion. Finally the free flight test. The MAV was held until it flapped at high frequency after which it was hand thrown in the forward direction. After it was thrown, the MAV continued to fly forward while slowly pitching upwards. The left and right controls were tested and the MAV showed that it could maneuver left and right. The last test was the pitching control. Increasing the rpm of the motor pitches the

MAV upwards and decreasing the rpm pitches the MAV downwards. However decreasing the rpm too much would make the MAV stall. This happened when the MAV tried to land. It stalled and crashed into the ground, breaking the parts that was holding the body. The flapping mechanism however is still intact. Although the MAV crashed during its landing, the test was a success, the MAV showed that it can fly and was able to be controlled remotely. The video clip can be seen at Youtube website: "<http://youtu.be/hp-Kpw6sl10>".



Fig. 15: Assembled Prototype 2.3

4. CONCLUSION

The objectives which were to design, build and fly a flapping wing MAV, was all met. The design section of prototype 1 and 2 has discussed and evaluated the conceptual designs of the flapping wing MAV. There were 2 fabrication methods that were used, laser cutting and 3D printing. Although it seemed that the 3D printing was a better fabrication method as it allows for more complicated design it does have its limitations in the area of melting point and breaking strength.

References

The references are shown in alphabetical order.

Component	Weight (g)	Parameter	Value
Brushless Outrunner Motor	20	Flapping Amplitude (deg)	50
Radio Receiver	11.5	Flapping Frequency (Hz)	10
Servos	4.5	Lift Coefficient	0.8
Li-Po Battery	4	Air Density (kg/m ³)	1.225
Electronic Speed Controller	10	Chord Length (m)	0.1
MAV Design	14.77	WingSpan (m)	0.15
Total Weight	64.77	Lift (g)	97.77

- [1] 2 Channel RC Ornithopter - Image of Cybird 2 2014.- [ONLINE]
Available at:

- <http://www.hobbytron.com/Cybird2ChannelElectricRCOrnithopter.html>
- [2] Alexander D. E., *Nature's Flyers, Birds, Insects and the Biomechanics of Flight*, The Johns Hopkins University Press, 2004.
- [3] Bejgerowski W., Ananthanarayanan A., Mueller D., and Gupta S.K., *Integrated product and process design for a flapping wing drive-mechanism*. ASME Journal of Mechanical Design, 131, 2009.
- [4] Bird Flight. *Flapping Wings Birds Flight* [ONLINE] Available at: <http://www.ornithopter.org/birdflight/flap.shtml>
- [5] Chklovski T.: *Pointed Wing Tips at Low Reynolds Number* [ONLINE] Available at: <http://www.wfis.uni.lodz.pl/edu/Proposal.html>
- [6] *Cybird Suchergebnisse. 2014. Image of Transverse Shaft Design* [ONLINE] Available at: <http://www.onemillionvideos.de/search?q=Cybird>
- [7] de Croon G.C.H.E., Groen M.A., De Wagter C., Remes B.D.W., Ruijsink R., and van Oudheusden B.W., *Design, aerodynamics, and autonomy of the Delfly*. Bioinspiration and Biomimetics, 7(2), 2012.
- [8] de Laurier, J. D., *The Development of a Full Scale Ornithopter Wing*, The Aeronautical Journal of the Royal Aeronautical Society, May 1993.
- [9] Ellington, C. P., *The Novel Aerodynamics of Insect Flight: Application to Micro- Air-Vehicles*, Journal of Experimental Biology, Vol. 202, No. 23, pp. 3439-3448, Dec. 1999.
- [10] Festo: SmartBird 2011, *Image of SmartBird ornithopter inspired by herring gull*. [ONLINE] Available at: http://www.festo.com/cms/en_corp/11369.htm
- [11] *Flapping Flight - How Birds Fly. Flapping Flight - How Birds Fly*. [ONLINE] Available at: <http://www.n6iap.com/ornithopter/howbirdsfly.html>
- [12] Hu H., Kumar A. G., Abate G., Albertani R., *An experimental investigation on the aerodynamic performances of flexible membrane wings in flapping flight*, Aerospace Science and Technology, Volume 14, Issue 8, December 2010, Pages 575-586, ISSN 1270-9638, Image of Dual Crank Design.
- [13] *Image of Single Gear Crank Design - University of California Biomimetic Millisystems Lab Ornithopter Project. 2014*. [ONLINE] Available at: <http://robotics.eecs.berkeley.edu/~ronf/Ornithopter>
- [14] Keennon M., Klingebiel K., Won H., and Andriukov A., *Development of the nano-hummingbird: A tailless flapping wing micro air vehicle*, proceedings of the 50th AIAA Aerospace Science Meeting, pages 6-12, 2012.
- [15] Krashanitsa, R. Y., Silin, D., Shkarayev, S. V., & Abate, G. (2009). *Flight dynamics of a flapping-wing air vehicle*. International Journal of Micro Air Vehicles, 1(1), 35-49.
- [16] Manuel N. : *Design of a Flapping Wing Mechanism*. Semester Project, Autonomous Systems Lab (ASL), Swiss Federal Institute of Technology Zurich (ETH), 2009.
- [17] McMichael, J., Francis, M.: *Micro Air Vehicles - Toward a New Dimension in Flight*, Defense Advanced Research Projects Agency, 1997.
- [18] *Robotic insects make first controlled flight*, Image of Robobee | Harvard Gazette. [ONLINE] Available at: <http://news.harvard.edu/gazette/story/2013/05/robotic-insects-make-first-controlled-flight/>
- [19] Shyy, W., Berg, M., and Ljungqvist, D., *Flapping and Flexible Wings for Biological and Micro Air Vehicles*, Progress in Aerospace Sciences, Vol. 35, No. 5, 1999, pp. 455-506.
- [20] Tennekes, H., *"The Simple Science of Flight: From Insects to Jumbo Jets"*, MIT Press, 4th edition, 2000.
- [21] Thomas J. Mueller: *International Journal of Micro Air Vehicles - On the Birth of Micro Air Vehicles*, Volume - Volume 1
- [22] Withers P. C.: *An Aerodynamic Analysis of Bird Wings as Fixed Aerofoils*, Portland State University Biology Research Paper, 143-162, 1981.

The Next Generation Design of Autonomous MAV Flight Control System SmartAP

Kirill Shilov*

Department of Aeromechanics and Flight Engineering
Moscow Institute of Physics and Technology
16 Gagarina st, Zhukovsky, Moscow reg., Russia 140180

ABSTRACT

The present article contains the most recent and significant results of the author's work in the field of autonomous flight control system for multirotors. The work is based on author's custom designed autopilot called SmartAP. By the moment, the system has two generations - version 1.0 released in 2012 and version 2.0 released recently in 2014. The changes affected both hardware and software. The new architecture is capable of running more complex algorithms helping to increase navigation precision and enhancing autonomy level. The developed system is capable of fully autonomous flight including take off and landing.

1 INTRODUCTION

Aerial monitoring has always been important part of surveillance and terrain exploration. Until recently, this task could have been solved only with piloted aircrafts, however, due to the latest progress in electronics, becomes possible creating of small Unmanned Aerial Vehicles (UAV) or Micro Air Vehicles (MAV). The capability of carrying payload (photo / video camera) and accomplishing preprogrammed missions autonomously helps to reduce exploitation costs significantly and doesn't require piloting skills from operator. Special interest is paid to multirotor UAVs which are capable of vertical take off and landing and hovering flight. These features makes them easy to operate in indoor environment and space-limited conditions. Obviously, the behavior of such vehicles is unstable and onboard Flight Control System (FCS) responsible for stabilization (the ability to stay in the air) and navigation (the ability to follow the desired trajectory) is required.

Research and development in the area of autonomous multirotor flight control systems are of interest. This paper contains the information on the difference in hardware design between the first and the second generation of the author's custom developed system as well as the description of the software architecture, navigation and control algorithms and provides the results achieved during autonomous flight tests.

*Email address: aviaks.kirill@gmail.com



Figure 1: Quadrotor powered by SmartAP autopilot performs the mission at IMAV 2013, France

2 THE HARDWARE

One of the main requirements for the next generation hardware were compact size and maximum compatibility with the previous version. The routing was done on four-layer PCB, the components placed on both sides. The size of the board is 60x40 mm, weight is 16 grams. The top side of the board is shown in the Figure 2.

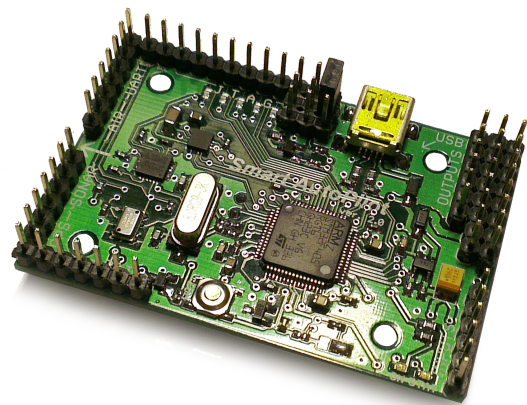


Figure 2: SmartAP v2.0 Hardware

This following section explains the general differences in the hardware design between two generations of the board:

2.1 Microcontroller

One of the key features of the second generation hardware is more powerful microcontroller. The first generation of the hardware has STM32F103 microcontroller, while the next is based on STM32F405 [1]. It has not only higher operating frequency (168 MHz vs 72 MHz) and more flash memory (1 MB vs 256 kB), but also Floating Point Unit (FPU) module which helps to reduce CPU load significantly during math operations, therefore, allowing to perform more complex attitude estimation algorithms and adaptive control loops.

2.2 Pressure sensor

Pressure sensor (barometer) plays very important role in MAV's altitude estimation and defines the precision of altitude hold mode. The previous generation of the board has Bosch BMP085 [2] barometer. The best accuracy of the measurements with this sensor was around ± 80 cm. After studying and comparing various pressure sensors test results it was concluded that MS5611-01BA03 [3] barometer from Measurement Specialties has the best performance available at the moment. The typical measurement results for two sensors are shown in the Figure 3 and 4 below:

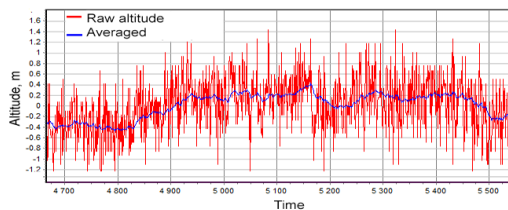


Figure 3: BMP085 pressure measurement results

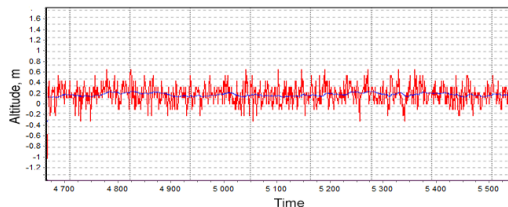


Figure 4: MS5611 pressure measurement results

2.3 Signaling

Sound and light signaling devices can be used to inform a user about the state of MAV (arm / disarm, calibration procedure, error and etc.). Therefore, two external digital output ports triggered through transistor for electromagnetic sounder and bright LED (maximum current is 100 mA for each channel) were added.

2.4 Battery voltage monitoring

One of ADC input ports connected through voltage divider allows measuring battery voltage directly in the range up to 30 V with 12 bit resolution.

2.5 External ports

Several external power supply ports (3.3 V and 5 V) as well as I2C line were added for further sensors on breakout board tests and external magnetometer connection [4].

3 SOFTWARE ARCHITECTURE

Autonomous control of multirotor MAV can be achieved with the ability of maintaining desired orientation and position in space, therefore, FCS is subdivided into three parts:

1. Information acquisition from the sensors (accelerations, angular rates, GPS position, atmospheric pressure and etc.)
2. Obtained information is exposed to verification, filtering and fusion in order to improve accuracy
3. Navigation and control algorithms execution.

The software architecture is based on Real Time Operating System FreeRTOS [5] running on STM32 ARM Cortex M4 microcontroller and presented in the form of a block diagram in the Figure 5:

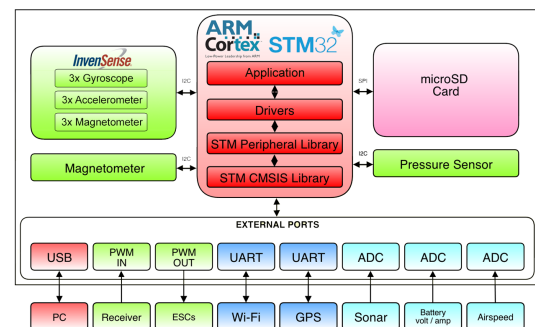


Figure 5: Block diagram architecture of the system

The software is divided into four subgroups. STM CMSIS Library and STM Peripheral Library form hardware abstraction level. The highest level - Application, includes attitude estimation and control algorithms as well as autopilot service functions. Therefore, the transition to another hardware can be done easily. The greatest computational time is taken by IMU attitude estimation algorithm and sensors' data acquisition task. Nevertheless, IDLE time is 71%. More details concerning task and corresponding CPU load are presented in the Table 1.

Task name	CPU Load, %
IDLE	71
IMU attitude estimation	16
Sensors' data acquisition	8
Control algorithms	3
Other	2

Table 1: Task name and corresponding CPU load

4 INERTIAL MEASUREMENT UNIT

The previous generation of SmartAP was based on InvenSense Digital Motion Processor (DMP), meaning that the sensor itself calculated attitude solution presented in the form of quaternions [6]. This method contributes to a significant reduction of main CPU load, however, it provides only 6 DoF solution since MPU-6050 [7] doesn't have integrated magnetometer and standalone Z-gyroscope bias correction is impossible.

Therefore, it was decided to refuse from DMP and read raw accelerometer, gyroscope and magnetometer data instead. Attitude estimation uses a quaternion representation, allowing accelerometer and magnetometer data to be used in an analytically derived and optimized gradient-descent algorithm to compute the direction of the gyroscope measurement error as a quaternion derivative.

This approach helps reducing measurement error related with gyroscope bias and also makes the solution more sustainable to vibrations. Attitude estimation with the described algorithm proved to be better than DMP solution in real tests. Screen of application for real-time IMU comparison tests is shown in the Figure 6.

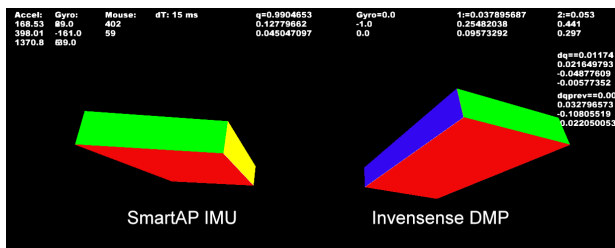


Figure 6: Application for real-time comparison of internal and DMP attitude solutions (left parallepiped - own solution, right parallepiped - DMP solution)

Yaw control of MAV was tested during the flight and showed good results for transition process when reference heading was changed as well as the absence of bias. The results are shown in the Figure 7.

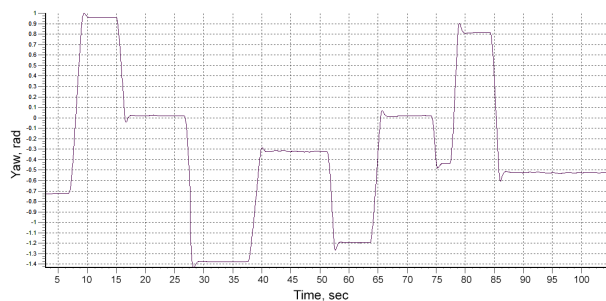


Figure 7: Measured yaw angle

5 CONTROL

The control algorithms are based on PID controller [8], which output signal $u(t)$ can be presented as follows:

$$u(t) = P + I + D = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

where $e(t)$ is an error and K_p , K_i and K_d are proportional, integral and derivative gains respectively. This method can be applied for angles stabilization, as shown in the Figure 8 below:

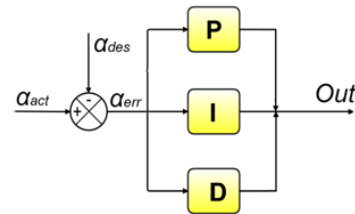


Figure 8: PID control block diagram

where α_{act} is measured angle and α_{des} is a reference angle. However, this approach turned out to be quite difficult for tuning and satisfied process behavior couldn't been achieved. Typical response curve for PID loop angle control looks as shown in the Figure 9:

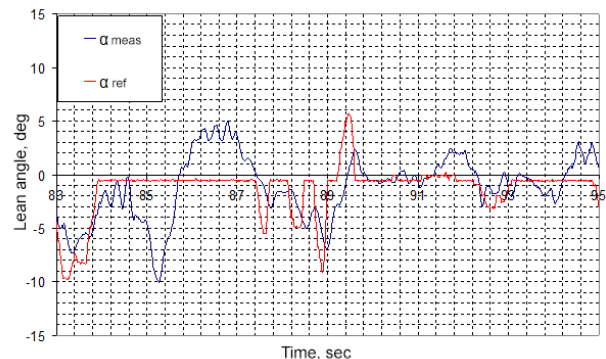


Figure 9: Response curve for PID loop

The same idea, but with some modifications gave much better results. It includes two PI controllers, one is used to stabilize the angular rate of rotation, while the second is used to stabilize the angle itself. This method turned out to be much easier for tuning and gave better results of vehicle's stabilization. Block diagram of the described method is presented in the Figure 10.

And typical response curve is shown in the Figure 11.

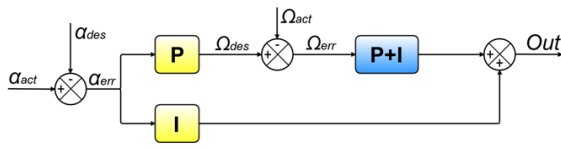


Figure 10: PI-PI control block diagram

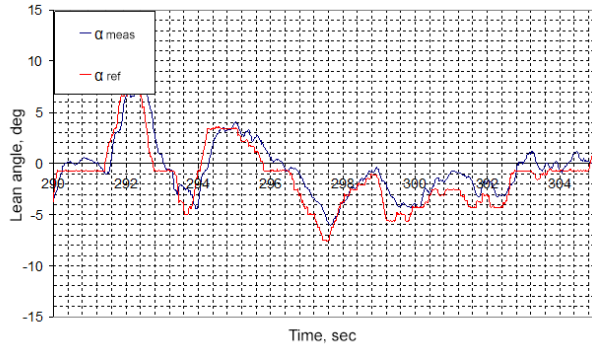


Figure 11: Response curve for PI-PI loop

6 AUTONOMOUS TAKE OFF AND LANDING

Take off and landing modes are ones of the most important for autonomous flight and had been studied and tested carefully.

6.1 Take off mode

After activating takeoff mode the motors start spinning and MAV tries to achieve the predefined altitude of 5 meters. Upon reaching this altitude it goes to "Loiter" mode. Altitude can be measured with barometer and ultrasonic range finder (sonar). Sonar gives better resolution (1 cm), however, its measurement results are very noisy at low distances. Pressure sensor has much less resolution, but measurements are not affected by noise. One of the difficulties is determination of average throttle value required for hovering. This value is fixed when lift-off occurs. The typical take off pattern is shown in the Figure 12. Both barometer and sonar altitudes are presented, where sonar noise can be clearly seen during lift-off. After achieving the altitude of five meters, accuracy of altitude hold is no worse than ± 50 cm.

6.2 Landing mode

After activating landing mode MAV maintains the vertical speed of -0.5 m/s. Touch down with the ground is determined by Z-accelerator readings, which in turn signalize that the motors can be turned off. The typical pattern of the landing is shown in the Figure 13.

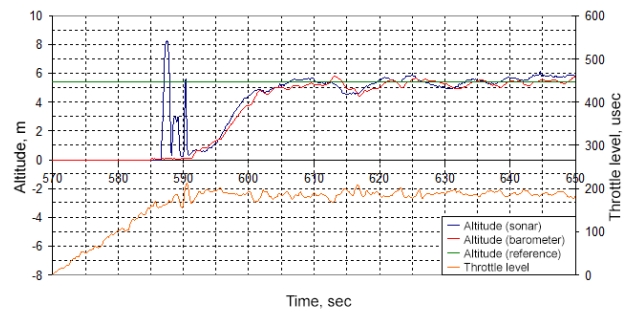


Figure 12: Flight parameters during autonomous take off

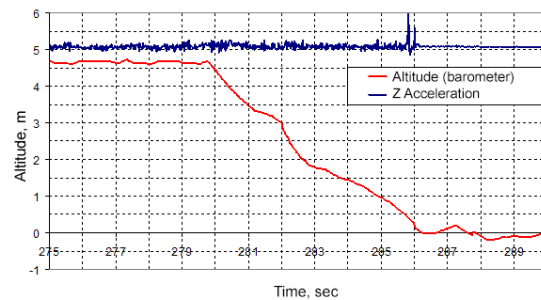


Figure 13: Flight parameters during autonomous landing

The vehicle is holding the altitude of 4.7 m. At 286 sec it is starting descending with the vertical speed of 0.7 m/s. Peak in acceleration readings can be clearly seen meaning the vehicle touched the ground. The system shuts down motors and disarms.

7 AUTONOMOUS POSITION HOLD

Similar PI-PI cascade is used for position control. Firstly, the reference velocity for current position error is calculated. Reference velocity is compared to the measured one and is converted to lean angles in north and east directions required to achieve the reference position. Navigation algorithm converts north / east lean angles to roll and pitch angles. The accuracy of position hold is about ± 0.5 m. The typical trajectory of the vehicle in position hold mode is presented in the Figure 14.

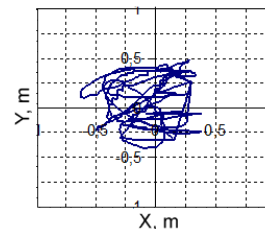


Figure 14: MAV Trajectory in position hold mode

8 AUTONOMOUS WAYPOINT FLIGHT

Autonomous waypoint flight mode is generally based on the position hold mode. Upon reaching every new desired point, the next point of waypoints list is set as the desired until the last waypoint is reached. After that, home point is set as the desired.

Reference trajectory (including four waypoint and home position) and vehicle's trajectory are shown in the Figure 15:

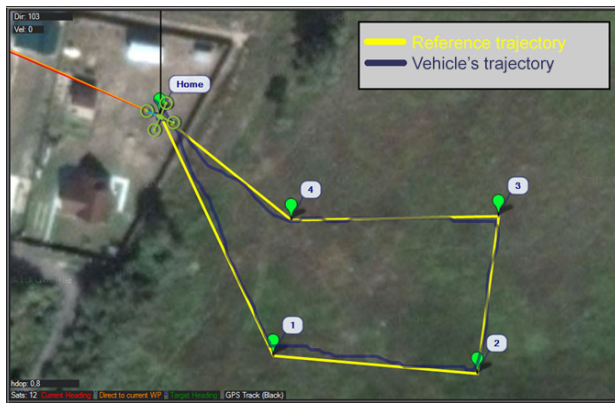


Figure 15: MAV trajectory in waypoint flight

9 GROUND CONTROL STATION



Figure 16: Android-based Ground Control Station

Communication between MAV and operator can be done with the help of the Ground Control Station (GCS) through the radio link. The autopilot supports MAVLink protocol [9], meaning that it is compatible with any GCS supporting MAVLink too.

GCS provides high-level mission control and doesn't require piloting skills from operator, which makes the system very easy to use.

One of supported ground stations is Droidplanner (for Android-based devices) is shown in the Figure 16.

10 CONCLUSION

The general aspects regarding multirotor MAV flight control system design for autonomous flights were considered in the present article. Brief overview of the hardware and software architectures was given as well as the differences between two generations of the board. Two attitude estimation methods had been compared. Two types of PID controllers had been also tested and compared. Autonomous take off and landing mode based on barometer and ultrasonic range finder altitude estimation had been analyzed. Some results on autonomous waypoint flight presented as well. Current research topics are positioning accuracy improvement by the means of multisensor data fusion and Kalman filtering and model-based adaptive control algorithms.

More information can be found on project's website [10].

11 REFERENCING

- [1] High-density performance line ARM-based 32-bit MCU, USB, CAN, 11 timers, 3 ADCs, 13 communication interfaces, STMicroelectronics, <http://st.com/>
- [2] BMP085, Digital, barometric pressure sensor, BOSCH, <http://www.bosch-sensortec.com>
- [3] MS5611-01BA03 Barometric Pressure Sensor, with stainless steel cap. DA5611-01BA03_011 www.meas-spec.com. Oct. 26, 2012. 000056111624 ECN1742
- [4] 3-Axis Digital Compass IC HMC5883, Honeywell, <http://honeywell.com/>
- [5] FreeRTOS - market leading real time operating system (or RTOS) from Real Time Engineers Ltd., <http://www.freertos.org/>
- [6] Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality, J. B. Kuipers, 2002
- [7] MPU-6000 and MPU-6050 Product Specification Revision 3.2, Invensense, <http://invensense.com/>
- [8] Introduction to PID Controllers - Theory, Tuning and Application to Frontier Areas, Edited by Rames C. Panda, ISBN 978-953-307-927-1, 258 pages, Publisher: InTech, Chapters published
- [9] MAVLink Micro Air Vehicle Communication Protocol - <http://qgroundcontrol.org/mavlink/start>
- [10] <http://sky-drones.com/>

Automatic extraction of moving objects from UAV-borne monocular images using multi-view geometric constraints

M. Kimura*, R. Shibasaki, X. Shao, and M. Nagai
University of Tokyo, Tokyo, Japan

ABSTRACT

This paper proposes a method to detect dynamic objects in the images obtained by a small UAV. Two geometric constraints in multi-view images are used to classify each of the extracted feature-points as static or dynamic. The first constraint is the epipolar constraint which requires static points to lie on the corresponding epipolar lines in the subsequent image. The second constraint, named as flow-vector bound constraint here, restricts the motion of static points along the epipolar lines. In addition, the pose of the UAV-borne camera, which is required when applying these constraints, is estimated by using a vision-based SLAM method, PTAM. The proposed method fully exploits the characteristics of UAV-borne images and achieves satisfactory results. The algorithms were tested with a small quadrotor platform in a real-world scene and successfully detected features extracted from multiple pedestrians.

1 INTRODUCTION

Detection of dynamic objects from images has been widely studied in computer vision research for many applications, such as traffic supervision, robot navigation, and crowd surveillance. This paper primarily focuses on moving object detection from the images obtained by small unmanned aerial vehicles (UAVs). It is not easy to detect dynamic objects from moving cameras since there are two motions involved: the motion of moving objects and the motion of the camera itself. Dynamic object detection from small UAVs is especially challenging because of the characteristics of these vehicles, such as continuous unrestricted pose variation and bad vibrations. To address these characteristics, new approaches are needed.

Jung *et al.* applied a probabilistic approach to detect moving objects from a mobile robot using a single camera in outdoor environments[1]. The changes in the images caused by camera motion is compensated using corresponding feature sets and outlier detection, and the positions of moving objects are estimated using an adaptive particle filter and EM

algorithm. Their algorithms were also tested with unmanned helicopter. Rodriguez *et al.* developed a real-time method to detect and track moving objects from UAVs using a single camera[2]. The main concept proposed in their work is to create an artificial optical flow field using the camera motion between two subsequent images. They compare this artificial flow with the real optical flow directly calculated from the images to detect features that belong to dynamic objects. Siam *et al.* proposed a automatic multiple moving target detection and tracking framework that executes in real-time and is suitable for UAV imagery[3]. Their framework is based on image feature processing and projective geometry, homography. The outlier image features, which violate homography, are computed with least median square estimation and clustered spatially as dynamic objects. These dynamic objects are tracked using Kalman filtering while persistency check is carried out to remove false detections.

These earlier studies[1, 2, 3] for moving object detection from moving platforms including UAVs focused on how to discriminate the changes in image sequences caused by dynamic objects from the ones caused by the camera motion. In other words, the motion of the platform is considered as a disadvantage for moving object detection in these approaches and their performances are thought to be best when the platform is not moving. In contrast, we propose an approach for moving object detection utilizing the motion of UAVs. As mentioned below, our approach uses multi-view geometric so that it can detect moving objects from UAV-borne images in real-time. There are some preceding studies using multi-view geometric constraints for moving object detection from moving platforms. Some of these studies are introduced below and we clarify the contribution and novelty of our approach.

Takeda *et al.* proposed a method to detect moving obstacles using the residual error calculated in the process of FOE (Focus Of Expansion) estimation[4]. At first in their method, the dense optical-flow field is extracted from sequence of dynamic images captured by a camera fixed on a moving platform. Next, the FOE is estimated in local image regions. An image region corresponding to the block is added with the residual error. This process is repeated by sliding and adding for the local region while changing the size of the local region. Finally, regions which have high residual error values are detected as candidate regions of moving obstacles. Experiments using ground-vehicles show that the method works

*Email address: motoki@iis.u-tokyo.ac.jp

well in a real outdoor scene. However, this method assumes the pure translation as the platform motion and the rotational motion is not assumed. Since small UAVs has unrestricted pose variation, thier method is not appropriate for UAV-borne images. Kang *et al.* developed an approach to detect and track independently moving regions in a 3D scene captured by a moving camera in the presence of the strong parallax[5]. Each of detected moving pixels are classified into independently moving regions or parallax regions by analyzing two geometric constraints, the commonly used epipolar constraint and the structure consistency constraint. Experiment results using airborne images show that their approach can successfully detect and track independently moving objects in a 3D scene despite of the strong parallax in the images. However, their approach is complex and unsuitable for the real-time process. Kundu *et al.* proposed a similar method to ours to detect moving object from image-sequences obtained by a robot on the ground[6]. Their approach uses two geometric constraints but it does not assume the rotational motion in one of the constraints. Besides, it requires other types of sensors, such as wheel encoders, to estimate the camera motion.

In this paper, we present an automatic method using multi-view geometric constraints to detect moving objects in UAV-borne images. This method fully exploits the unrestricted and continuous pose variation of UAVs and is appropriate for the small UAVs whose motion is unstable. The method can detect moving objects in real-time using an ordinal laptop computer and does not need sensors other than a monocular camera.

2 METHODOLOGY OVERVIEW

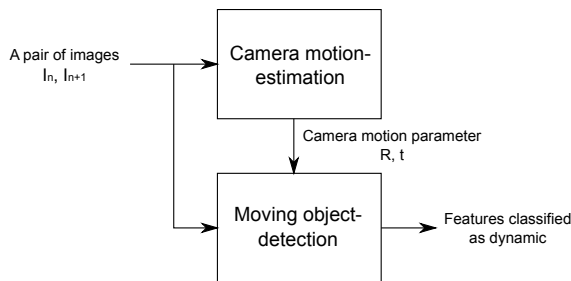


Figure 1: The overview of the proposed method

An approach proposed in this paper consists of two major parts shown in Figure 1. The first part is the one to estimate the camera motion equipped with the UAV. Using two images and a vision-based SLAM method, this part estimates the relative motion of the camera in a 3D scene between the two frames. This relative motion is the one so called camera extrinsic parameter, which includes the translation vector t and the rotational matrix R . These parameters are used in the second part, which is the one for moving object detection.

It uses geometric constraints in image sequences calculated from these parameters to detect feature-points which belong to independently moving objects in a 3D scene. The camera motion estimation part is detailed in section 3 and the moving object detection part is in section 4.

3 CAMERA MOTION ESTIMATION

In our approach, the camera extrinsic parameter between a pair of frames is estimated by a vision-based SLAM, PTAM (Parallel Tracking And Mapping)[7]. PTAM is a robust and real-time key-frame based SLAM mehod and have been applied to some vision based navigations for small UAVs[8, 9].

Using the image I_n captured at time-index n , PTAM calculates the camera pose z_n in a reference frame at n :

$$z_n = \begin{bmatrix} r_n \\ q_n \end{bmatrix}, \quad (1)$$

where r_n is the position and q_n is the the quaternion which describes the attitude of the camera. The camera extrinsic parameters or the relative motion between two captured images I_n and I_{n+1} can be calculated as being

$$t_{n:n+1} = -R(q_{n+1}) \times (r_{n+1} - r_n), \quad (2)$$

$$R_{n:n+1} = R(q_n^* \otimes q_{n+1}), \quad (3)$$

where $R(q)$ is a directional cosine matrix (DCM) defined by the quaternion q and q^* is the conjugation of the quaternion q . The symbol $\otimes$ represents quaternion products.

In the motion detection part, the calculated extrinsic parameters $t_{n:n+1}$ and $R_{n:n+1}$ are used to detect the moving objects in the images I_n and I_{n+1} .

4 MOVING OBJECT DETECTION

Figure 2 represents the process of the moving object detection part in our approach. An upper half of Figure 2 is the process of the moving object detection and lower half are the images which are the results of each step in the process. The details of each step in the process are explained below in this section.

4.1 Feature extraction and tracking

In the feature extraction step, sparse Kanade-Lucas-Tomasi (KLT) features[10] are extracted from images I_n and I_{n+1} captured at time indexes n and $n+1$. Next, in the feature tracking step, each of features extracted from the images are tracked by KLT feature tracker. Let p_n^i and p_{n+1}^i be the positions of the i th identical 3D-scene point X^i in images I_n and I_{n+1} , which are obtained by feature extraction and tracking steps. KLT features extracted from the image I_n are represented as red points in the left image in Figure 2 and tracking result of each feature between the images I_n and I_{n+1} is represented as green lines in the second image from the left in Figure 2.

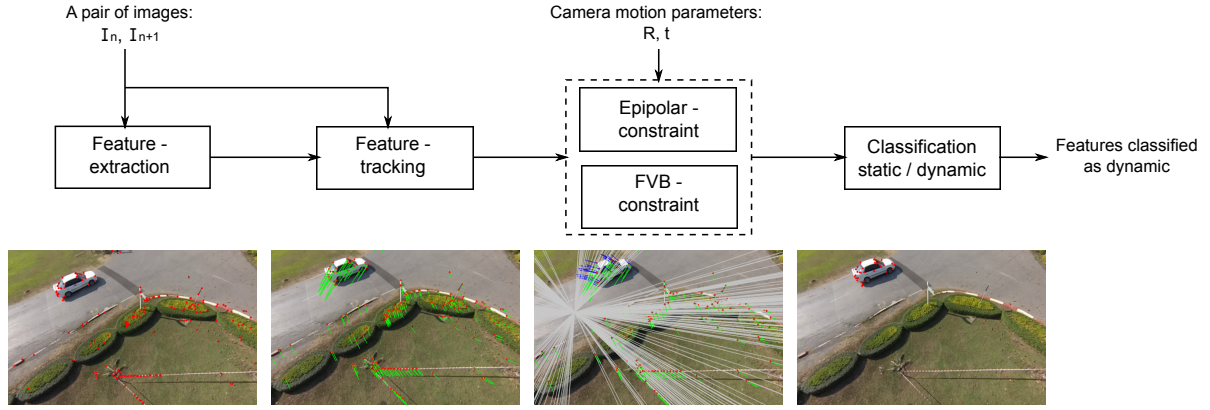


Figure 2: The process of moving object detection and result images of each step

Then, two geometric constraints are evaluated at each of features to classify each of them as static or dynamic. The first constraint we use is the epipolar constraint and the second constraint is the one called flow-vector bound (FVB) constraint in this paper. To calculate each constraint and detect dynamic feature points, the fundamental matrix between the images I_n and I_{n+1} is used. The fundamental matrix $F_{n,n+1}$ between the pair of images are defined as

$$F_{n,n+1} = K^{-T}[t_{n:n+1}]_{\times} R_{n:n+1} K^{-1}, \quad (4)$$

where K is the intrinsic matrix of the camera and $R_{n:n+1}$, $t_{n:n+1}$ is the rotation and translation of the camera between two views, which are given by the camera motion estimation part. The details of the constraints we use and how to evaluate them at each of features are explained in the rest of this section.

4.2 Epipolar constraint

The epipolar constraint is represented by $p_{n+1}^{iT} l_{n+1}^i = 0$, where l_{n+1}^i is the epipolar line in the image I_{n+1} corresponding to the feature p_n^i . The epipolar line l_{n+1}^i is given by: $l_{n+1}^i = F_{n,n+1} p_n^i$. This equation means that features which extracted from static point in a 3D scene to lie on the corresponding epipolar lines in the subsequent image. However, if a point is not static in a 3D scene, the feature p_{n+1}^i may be off the corresponding epipolar line l_{n+1}^i and the perpendicular distance from the feature to the epipolar line, $h_{epi,n+1}^i$ is not zero as shown in the left figure of Figure 3.

If the coefficients of the line l_{n+1}^i are normalized, the perpendicular distance in the image I_{n+1} is given by $h_{epi,n+1}^i = |l_{n+1}^i \cdot p_{n+1}^i|$. Similarly, the perpendicular distance in the image I_n is given by $h_{epi,n}^i = |l_n^i \cdot p_n^i|$. If the value of $h_{epi,n}$ or $h_{epi,n+1}$ is far from zero, it is more likely to be an image of the moving point.

The evaluation step of the epipolar constraint is shown in the second image from the right in Figure 2. The white lines

represent the epipolar lines and the blue lines represent the distance from features to the corresponding epipolar lines.

4.3 Flow-vector bound (FVB) constraint

When the camera does not move, the epipolar line cannot be defined. Besides, when the degenerate motion arises, moving points cannot be detected with the epipolar constraint since the features move along the epipolar lines even though they belong to dynamic points in a 3D scene as shown in the right figure of Figure 3.

We use the flow-vector bound constraint as the second constraint to detect moving points correctly during degenerate motions. Assuming the pin-hole camera model, we get the equation which describes the feature movement in the images:

$$p_{n+1}^i - K R_{n:n+1} K^{-1} p_n^i = \frac{1}{z} K t_{n:n+1}, \quad (5)$$

where z is the depth of a static 3D point corresponding to the features p_n^i and p_{n+1}^i . If we set z_{max} and z_{min} as the upper and lower bound on z , we then find image displacement bounds along the epipolar line, d_{min} and d_{max} , corresponding to z_{max} and z_{min} using Equation 5. If the image displacement $d^i = |p_{n+1}^i - K R_{n:n+1} K^{-1} p_n^i|$ does not lie between d_{min} and d_{max} , it is more likely to be a dynamic point.

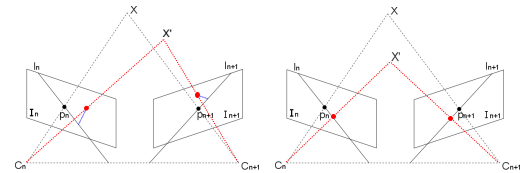


Figure 3: LEFT: a point X in a 3D scene moves non-degenerately hence its image point p does not lie on the corresponding epipolar line. RIGHT: The point X moves degenerately in the epipolar plane. Hence, despite moving, its image point p lies on the corresponding epipolar line.

4.4 Probabilistic model for the classification

As mentioned, we denote by p_n^i the i th feature p^i in the image I_n . The corresponding feature in I_{n+1} is denoted by p_{n+1}^i . The probability of p^i being stationary is defined as

$$P(p^i = \text{static}) = f_{EP} \times f_{FV}, \quad (6)$$

where f_{EP} and f_{FV} are defined as

$$f_{EP} = e^{-\alpha(|p_n^i \cdot t_n^i| + |p_{n+1}^i \cdot t_{n+1}^i|)}, \quad (7)$$

$$f_{FV} = \left\{ 1 + \left(\frac{d^i - d_{mean}}{d_{range}} \right)^\beta \right\}^{-1}, \quad (8)$$

where $d_{mean} = \frac{d_{max} + d_{min}}{2}$, $d_{range} = \frac{d_{max} - d_{min}}{2}$.

α and β are smoothing factors. If the probability is below the threshold, the feature p^i is classified as a dynamic point.

The values of these parameters and threshold need to be adjusted because the optimal values depend on the situation, such as the flight height, the velocity of the moving target, the image resolution, etc. We adjusted these values using particular image sequences captured from the UAV before the experiments.

The features classified as dynamic are shown as red points in the right image in Figure 2. The features which belong to the moving car are correctly detected.

5 EXPERIMENTAL RESULT

The algorithms we propose were tested in a real-world scene with a quadrotor-type UAV, AR.Drone2.0, shown in Figure 4.



Figure 4: The UAV used for the experiment (AR.Drone 2.0)

The UAV flew at the height of 10 meters over a crossing where some pedestrians walked. The results of the experiment were shown in Figure 5.

Extracted KLT features in the first image are shown in the topmost image in Figure 5. At this stage, features are extracted from both dynamic objects (pedestrians) and static objects. In the second image from the top in Figure 5, red points represent the features in the second image and green lines show the result of KLT tracking between the first image and the second image. The features extracted from both dynamic and static objects move in the image. In the third

image from the top in Figure 5, each white line is the epipolar line of the corresponding feature and each blue line represents the perpendicular line from feature to the corresponding epipolar line. Note that features extracted from pedestrians at right part of the image move vertically to the epipolar line, but features extracted from pedestrians at left part of the image move along the epipolar line. In the bottommost image in Figure 5, red points represent the points which classified as dynamic by our method. Only the features which belong to pedestrians are detected. Although some features extracted from pedestrians move along the epipolar line as can be seen in the third image from the top in Figure 5, those features are also detected by flow-vector bound constraint.

The proposed method was implemented using OpenCV library and could be run at maximally 15 fps on ordinal laptop computer (Intel Core i5-2540M, 2.6GHz, 4GB RAM). Computational resources are mainly consumed for the camera localization (PTAM).

6 CONCLUSION

We proposed a real-time method to detect the moving points from UAV-borne images using multi-view geometric constraints. The propose method makes the best use of the characteristics of small UAVs, such as their great mobility and pose variation. The algorithm was tested in a real-world scene, a crossing where some pedestrians walk, and the points which belong to dynamic objects were successfully detected.

As future works, we will develop a clustering method for grouping a set of moving points to a moving object and a robust tracking method to know the behavior of each moving object. We will also challenge the sensor fusion to estimate the camera motion. It improves the speed of our algorithm since vision-based localization is computationally expensive, and it enables us to estimate the motion of the camera robustly even in dynamic environments and texture-poor environments where the performance of PTAM is poor.

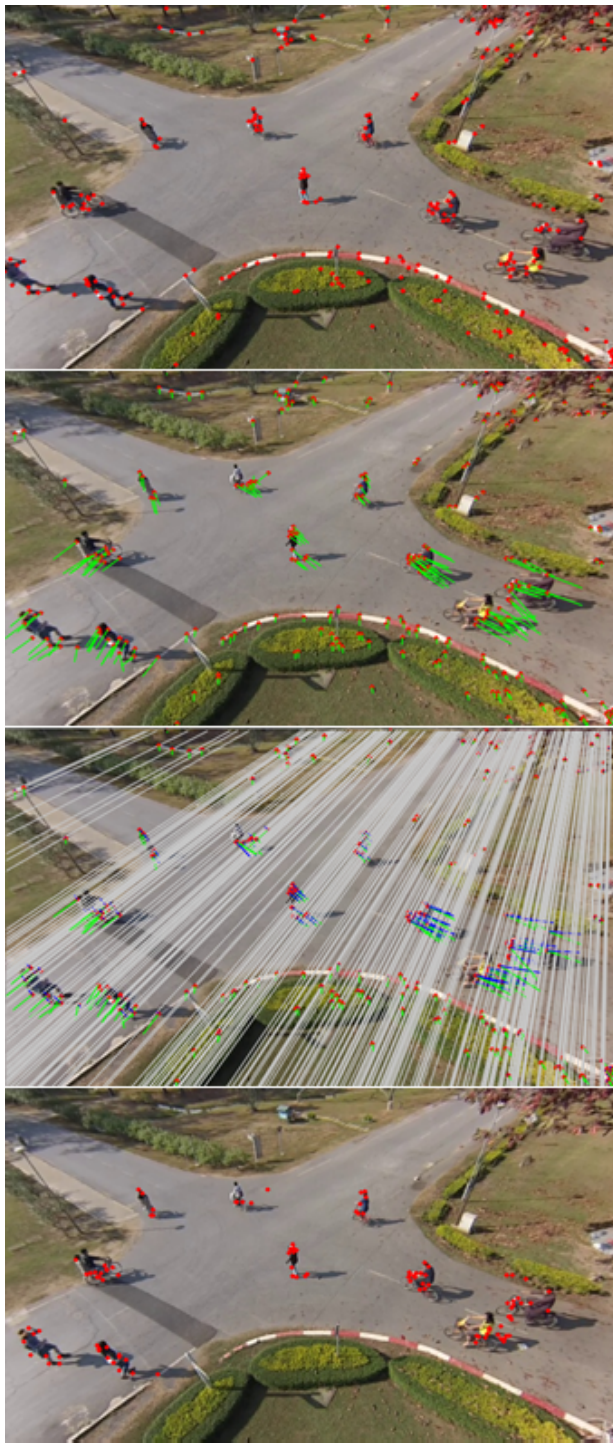


Figure 5: Topmost: Extracted features in the first image (red), The second from the top: Extracted features in the second image (red) and the result of feature tracking (green), The third from the top: Epipolar lines (white) and perpendicular lines from features to the corresponding epipolar lines (blue), Bottommost: Detected moving points (red)

REFERENCES

- [1] B. Jung and G. S. Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *International Conference on Intelligent Autonomous Systems*, pp. 980–987, 2004.
- [2] G. R. Rodriguez-Canosa, S. Thomas, J. del Cerro, A. Barrientos, and B. MacDonald. A real-time method to detect and track moving objects (DATMO) from unmanned aerial vehicles (UAVs) using a single camera. *Journal of Remote Sensing*, Vol. 4, No. 4, pp. 1090–1111, 2012.
- [3] M. Siam and M. ElHelw. Robust autonomous visual detection and tracking of moving targets in UAV imagery. *11th IEEE International Conference on Signal Processing (ICSP)*, Vol. 2, pp. 1060–1066, 2012.
- [4] N. Takeda, M. Watanabe, and K. Onoguchi. Moving obstacle detection using residual error of FOE estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vol. 3, pp. 1642–1647, 1996.
- [5] J. Kang, I. Cohen, G. Medioni, and C. Yuan. Detection and tracking of moving objects from a moving platform in presence of strong parallax. In *10th IEEE International Conference on Computer Vision (ICCV)*, Vol. 1, pp. 10–17, 2005.
- [6] A. Kundu, K. Krishna, and J. Sivaswamy. Moving object detection by multi-view geometric techniques from a single camera mounted robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4306–4312, 2009.
- [7] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, 2007.
- [8] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time Onboard Visual-Inertial State Estimation and Self-Calibration of MAVs in Unknown Environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 957–964, 2012.
- [9] J. Engel, J. Sturm, and D. Cremers. Camera-based navigation of a low-cost quadcopter, In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2815–2821, 2012.
- [10] D. B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *the 7th International Joint Conference on Artificial Intelligence*, Vol. 81, pp. 674–697, 1981.

Design of Ducted Fans of Small Height for Hexacopter with Long Hover Ability

Sergey V. Serokhvostov,*N. D. Ageev, M. E. Arkhipov
Moscow Institute of Physics and Technology
Department of Aeromechanics and Flight Engineering
140180, Gagarina street, 16, Zhukovsky, Russia

ABSTRACT

Investigated is the ability of the usage of small height ducted fans in multicopters for increase in the hover time and protecting the propellers and surrounding. A set of numerical and experimental investigations concerning duct aerodynamics and efficiency of electrical part (drive, speed controller, accumulator) was conducted. The effects of interference between two ducted fans situated close to each other are also investigated.

1 INTRODUCTION

Multicopters are promising aircraft for the set of applications. The main feature of such a remotely piloted aircraft (RPA) is the ability to hover for a long time over some point. From this point of view one of the main multicopter characteristics is hover time. It implies that the energy consumption from the onboard energy source must be minimized. For a number of reasons, electrical drives are used in multicopters and the energy source is the accumulator.

The energy consumption minimization implies the efficiency maximization of each part of powerplant and a powerplant as whole.

For the set of applications, the multicopter propellers as well as the surrounding environment must be protected from the contacts with obstacles and from rotating propellers, respectively. One of the solutions is to use the so-called "ducted fans".

On the one hand, the mass of the duct increases with the height increase. In addition, the high duct has worse performance in the presence of wind. On the other hand, with the duct fan height decrease the aerodynamical efficiency of the ducted fan decreases. The ducted fan must also fit the electrical part of the powerplant to provide the maximal efficiency.

The outer diameter of the duct was set as 17 inch for the investigation, and the thrust of the ducted fan was set as 9N.

2 NUMERICAL INVESTIGATION

First of all, the aerodynamics of the "short" duct was investigated numerically. For speeding-up the calculation process the propeller was modeled by "active disk" [1] and only

duct parameters were varied. The shape of the duct and disc position were optimized to provide maximal efficiency.

Considering the shape of the duct vertical cross-section as some kind of profile, the type of profile, its thickness and its angle of inclination with respect to the horizontal line (the plane of the active disc) were varied. As for the active disc, its position and gap between the disc and duct were varied.

The computational domain contains multiblock structured mesh. This type of the mesh provides the most reliable and detailed simulation. The size of the cells on the model surface doesn't exceed 1 mm. The general maximum cell size is equal to 1 m. The cell size growth ratio is 1.1, which provides sufficiently detailed simulation. The total 2D grid has 287 thousands elements, that is equal to 34 million in 3D case. The wall cell thickness is about 0.007 mm. The prism layer provides $Y_{+max} \sim 0.53$.

Figure 1 shows the mesh used.

The boundary conditions on the walls of the computational domain are standard: all the walls besides the duct and active disk are free slip walls. Duct is no slip wall and active disk is the interface with the pressure drop of 66.9 Pa. RANS system of governing equations with the SST turbulence model [2] was solved by the solver ANSYS CFXTM. Solving is carried out with an incompressible viscous fluid model with parameters corresponding to air at 25C.

The problem is solved in a stationary regime with the approximation scheme of the second-order. One calculation takes 1000 iterations in average. The stop condition is fixation of the third significant digit in magnitude of the aerodynamic forces acting on the model.

Figure 2 shows one of the best duct shapes found, the corresponding location of the active disc and the velocity distribution for the case investigated.

It was found that the main peculiarity affecting the flow near the duct is the vortex on the end of the active disc which can be seen in Figure 2 near the end of disc. It was also proved that the smaller the chink between the disc and duct, the better is flow near the duct.

The second peculiarity is that in contrast to the "long" duct, in the case investigated the inner surface of the duct is convergent (in the case of long duct the inner surface is convergent in the fore-part and divergent in the rear part).

It was also found that for the fixed outer diameter of duct the maximal obtained total efficiency of the "duct+disc" is

*Email address: serokhvostov@mail.ru

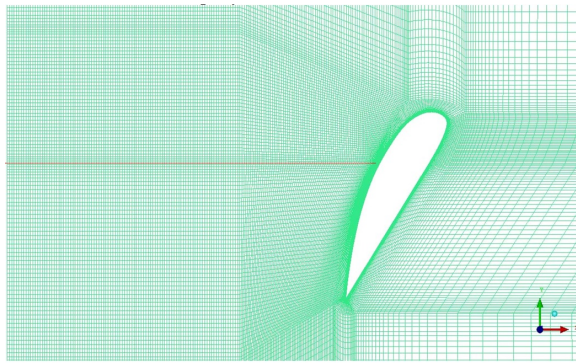


Figure 1: Computational mesh.

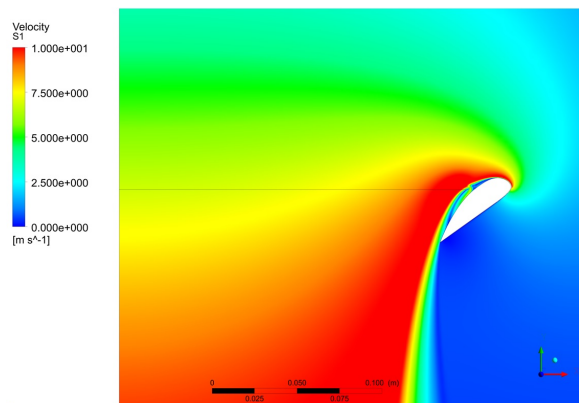


Figure 2: Velocity distribution near the duct for the case investigated.

nearly the same as for the disc with the diameter equal to the outer diameter of the duct.

3 EXPERIMENTAL SETUP

On the basis of the mathematical model of the duct the experimental test duct was made with the help of 3D-printing technology, see Figure 3. In addition, a “thin” duct was made as the “inner” part of optimized duct (in this case the “experimental test duct” was used as mould), see Figure 4. The weight of the “thin duct” is about 45–50 gram. Then the experimental setup was designed and made to measure the characteristics of the ducted fan, drive, speed controller and accumulators which enables to measure propeller rotational frequency, thrust, current from accumulator, voltage at the input of the speed controller.

4 EXPERIMENTS AND RESULTS

All the necessary data were obtained to calculate the total efficiency of a system and to define the efficiencies of its components. Experiments show that within the accuracy of the measurements there is practically no difference in the efficiency between the ducted fan and the propeller of the same

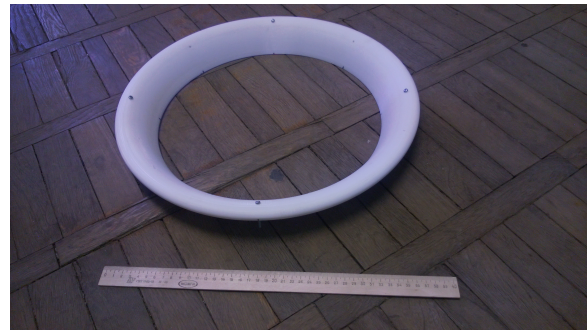


Figure 3: Duct made by 3D-printing technology.



Figure 4: Thin duct.

diameter. However, it should be mentioned that the authors are not sure that the elements of the ducted fan (propeller, fan and their combination) really have the best efficiencies among all the possible variants.

The interference influence of the two ducted fans was also investigated. The situation was modeled with the help of the “wall” situated in the vicinity of the duct. It was found that within the experiment accuracy there is no effect of interference.

After that, a set of experiments was conducted to define the dependence of the speed controller efficiency on its maximal current and voltage permitted and voltage of power source. A set of controllers (see Figure 5, Table 1) were tested for the case of 9N thrust of the powerplant for the different input voltages (12–18 Volt). First, it was found that for the current much more lower the maximal permitted (less than 30%) the controller efficiency is practically independent of the value of the maximal permitted current. This means that one can utilize the controllers with maximal permitted current of about 2–3 times higher than nominal current.

Second, the characteristic of the maximal permitted voltage does not affect the efficiency of the controller (at least for low current). Third, the efficiency of the controller decreases with voltage increase. For example, the increase of the voltage from 12V to 18V gives the increase of the current of about 10%. Finally, it was found that the mass of speed con-

	Max. current, A	Max voltage, Li-Po
Jeti Advance 40 Plus	40	6
Multiplex Multicontrol 40	40	6
ZTW 40A	40	6
Hornet	60	6
Hobbywing Flyfun 60A	60	6
Maytech 50	50	6
T-Motor T40A 400Hz	40	6
Scorpion 45	45	3
Scorpion 35	35	3

Table 1: Controllers tested.

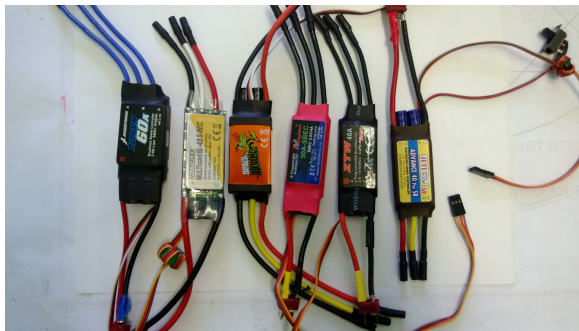


Figure 5: Controllers tested.

troller is proportional to the maximal permitted current and practically independent of the maximum voltage. The coefficient of proportionality is roughly about 1 gram per 1 Amper.

5 CONCLUSION

1. Numerical investigations of small height ducted fan were conducted. Optimal duct shape and propeller position were found. It was obtained that the efficiency of ducted fan is practically the same as for the propeller of the same diameter.
2. Experimental setup was made. All necessary characteristics of ducted fan were obtained. It was found that the efficiency of ducted fan is nearly the same as for the propeller of the same diameter.
3. The characteristics of speed controllers were investigated. The dependence of controller efficiency on the maximal permitted voltage and current and input voltage were estimated.

REFERENCES

- [1] H. Glauert. *The elements of aerofoil and airscrew theory*. Cambridge Science Classics Series. Cambridge University Press, 2 edition, 1983.

- [2] F. R. Menter. Zonal two equation $k-\omega$ turbulence models for aerodynamic flows. *AIAA Paper 93-2906*, 1993.

Design and Build of Swarm Quadrotor UAVs at UGS

Alvin Ng, Qiwang Foo, Ridhwan Jumat, Tony Rayar Victor, Wei Chong Koh, Sutthiphong Srigrarom
University of Glasgow Singapore

Contact e-mail: spot.srigrarom@glasgow.ac.uk

ABSTRACT

At the University of Glasgow, we have been developing the swarm of autonomous micro-unmanned aerial vehicle (μ -UAVs), as a part of team projects of the final year project design course: Aerospace Systems Design Project 4 for our BEng in Aerospace Systems program. For our Singapore campus, where similar course is conducted, the platform chosen is the quadrotor. In this paper, we present the conceptual design, construction, flight test, swarm behaviour and lessons learnt.

1. INTRODUCTION

A Quad-rotor consists of four motors with a rotor attached to each. When the rotor spins, it generates lift vertically. Lift increases when the rotor spins faster and vice versa. Hence Quad-rotor's climb and descent is dependent on the speed of the rotor. In order to hover, the sum of lift produced by the 4 rotors must be equivalent to the gravitational force.

Manoeuvring along the horizontal plane is achieved by tilting the Quad-rotor in the desired direction. The difference in the Lift between the rotors causes tilting. The horizontal component of lift results in movement along horizontal plane.

If all rotors spin in clockwise direction, the Quad-rotor will spin undesirably in the opposite direction. To prevent this the alternate rotors spin in opposite direction. Therefore the effect of the torque generated by the two opposing pairs of spinning rotors is balanced when all rotors are rotating with equal speed. In Quad-rotor flight, the moment of the vehicle is determined by its velocity, angle of attack, and its airframe. It has to be considered because they affect the attitude control, and thrust variation of the rotorcraft.

2. QUADROTOR MODELLING

2.1 Frame of references

The Quad-rotor is modelled in a Flat Earth coordinate reference. Flat Earth reference neglects the centrifugal and Coriolis accelerations since the trajectory of vehicle is much slower than the orbital speed. This section discusses about the dynamics equations of motion of the Quadrotor and how it was transformed into the Flat Earth Frame of Reference.

Frame of reference denoted as (x, y, z) axes is simply an axes system which is used to define quantities or measurements. The axes can be fixed to a body or earth. In this case, the body axes (x_b, y_b, z_b) is rotating with respect to the centre of gravity of the

quad-rotor and the earth axes (x_e, y_e, z_e) is with respect to the earth as shown on Fig 1.

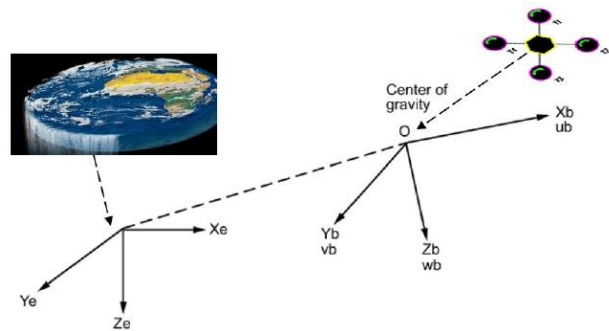


Fig. 1 Flat Earth Reference Frame

Relationship between the quad-rotor's frame of reference and the earth's frame of reference can be expressed by a transformation matrix as shown below. Such matrix is obtained using the three Euler angles commonly used in an aircraft in a prescribed sequence of rotation of roll, pitch and yaw.

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ -c\phi s\psi + s\phi s\theta s\psi & -c\phi c\psi + s\phi s\theta c\psi & s\phi\theta \\ s\phi s\psi + c\phi s\theta s\psi & s\phi c\psi + c\phi s\theta c\psi & c\phi\theta \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix}$$

where s, c denote sine and cosine functions respectively.

2.2 Equations of motion

As forces are produced by the total thrust of the rotor in the negative z direction and gravity is acting on the quad-rotor in the positive z direction, to transform forces from the body frame of reference to the earth frame of reference, the transformation matrix is used and therefore resulting in the following translational equations

$$\begin{aligned} \ddot{x} &= -\frac{U}{m} (\cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi) \\ \ddot{y} &= -\frac{U}{m} (\sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi) \\ \ddot{z} &= -\frac{U}{m} (\cos\theta \cos\phi) + g \end{aligned}$$

For rotational motion, J is the inertia matrix which is calculated by assuming rotors as individual point mass located

at a distance away from the quad-rotor's central hub. The hub is considered as a dense sphere with radius and mass at the center as shown on Fig. 2.

Therefore,

$$J_x = J_y = \frac{2MR^2}{5} + 2ml^2 \text{ and } J_z = \frac{2MR^2}{5} + 4ml^2$$

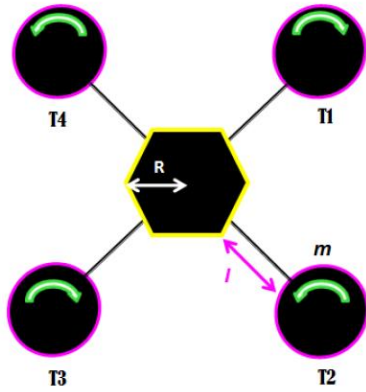


Fig. 2 Quad rotor dimensions

The equations of rotational motion are

$$\ddot{\phi} = \frac{1}{J_x} \tau_{\phi} \quad \ddot{\theta} = \frac{1}{J_y} \tau_{\theta} \quad \ddot{\psi} = \frac{1}{J_z} \tau_{\psi}$$

2.3 Dynamics of motor

Since thrust for the quad-rotor are produced by the rotors, it is important to consider the dynamic of the motor when modelling the quad-rotor. The DC-motor is able to convert electrical energy to mechanical energy. With a given voltage input, the rotational speed of a motor can be calculated as shown below.

$$V_{in} = R_m i + L_m \frac{di}{dt} + k_e \dot{\Omega} \text{ where } \dot{\Omega} = \frac{k_m i}{J_m}$$

By integrating $\dot{\Omega}$ to obtain Ω , the thrust generated can be calculated using the following set of equations.

$$C_t = \frac{T}{\rho n^2 D^4} \text{ where } n = \frac{\Omega}{2\pi}$$

Therefore,

$$T = C_t \rho \left(\frac{\Omega}{2\pi} \right)^2 D^4$$

Finally, with the calculated thrust and measured, following flight performance can be determined.

$$U_1 = T_1 + T_2 + T_3 + T_4 \quad (\text{hover})$$

$$\tau_{\phi} = l[(T_3 + T_4) - (T_1 + T_2)] \quad (\text{roll})$$

$$\tau_{\theta} = l[(T_1 + T_4) - (T_2 + T_3)] \quad (\text{pitch})$$

$$\tau_{\psi} = (T_1 + T_3) - (T_2 + T_4) \quad (\text{yaw})$$

3. QUADROTOR CONSTRUCTION

3.1 Structure

The Quad-Rotor platform was bought by UGs and given to us to assemble. The package comes with pieces of the quadrotor. The main structural components of our quad-rotor are summarized on the following table.



Fig. 3 Completed Structure of Quad-Rotor

Qty	Name	Dimension
1	Central Hub	155mm
4	Motor Arm	330mm
2	Stand	220mm

Central Hub

The central hub is made up of 2 carbon fibre sheets that are fastened together while leaving space for the motor arms to be secured to. The central hub also has an elevated platform for the placement of the ArduPilot Mega 2.5. The interior of the central hub holds all the wires such as the power module and the 4 to 1 conversion adapter. The Li-polymer battery is cable tied to the bottom of the central hub for better stability.

Motor Arm

The motor arms are made up of hollow carbon rod. These rods are securely fastened to the central hub using rubber spacers. This is to reduce the amount of vibrations caused by each motor. The rods are positioned to be equally spaced from each other. The motors are all seated on a carbon fibre sheets which are fastened together using bolts and nuts.

Stand

The stand is also made up of hollow carbon fibre rod. The part that is connected to the central hub is sandwiched between 2

carbon fibre sheets to strengthen the joints. The bottom stands are to provide more surface contact to the ground. It also allows the vehicle to have a more stable take off by enforcing it to be at an upright position.

3.2 Electronic Components

The electronic components used for our platform are summarized on the followings table:

Qty	Name	Weight
4	ESC	32g
4	Brushless Motor	55g
1	Li-polymer Battery	250g
4	Plugs	1g
1	4 to 1 Conversion Adapters	3g
1	Power Module	3g
1	ArduPilot Mega 2.5	31g
1	Receiver	9.4g

Electronic Speed Controller (ESC)

The Electronic Speed Controller (ESC) chosen, is specially designed for multi-rotor motor. Each ESC weights 32grams with the dimensions of 55mm by 26mm by 11.6mm (width by length by height). The brushless ESC provides variable power to the motor allowing proportional speed adjustments. This ESC also has a voltage cut-off function which lowers power when the voltage is low.

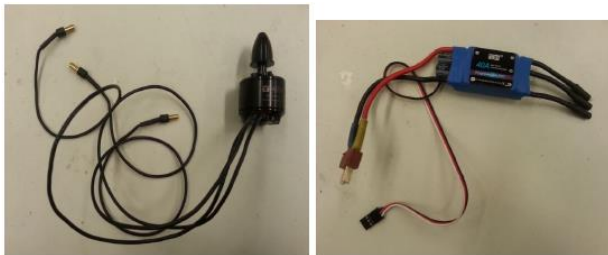


Fig. 4 Brushed ESC DYS OPTO-40A and Tiger Motor MT2212-13 - KV

Brushless Motor

The motors (Fig.4) used are out runner brushless. Outrunner motors have a lower speed however greater torque which allows us to direct drive a larger propeller without the use of a gearbox. Each motor weighs around 55grams with the dimensions of 27.5mm by 28mm (length by height) and can generate lift up to 800grams depending on the propeller used.

Batteries

The battery used is a 3 cell Lithium Polymer. It has a voltage value of 11.1V and current rating of 3300mAh. The higher the rating the longer it can provide power to the other electronic components. Since we are using 4 motors we require a strong battery however power has its price. The stronger the battery the heavier and bigger it is. For our case it sets us back to 250 grams and dimensions 43mm by 130mm by 13mm (width by length by height). The battery could last a maximum of around 10 to 15 minutes of flight time.



Fig. 5 RevoLectrix Li-polymer Battery

Plugs

The plugs are soldered to the output of the ESC to connect to the batteries which has a Dean plug female. It allows 40A current to pass through. This connector ensures secure connection while easy for plugging and unplugging.

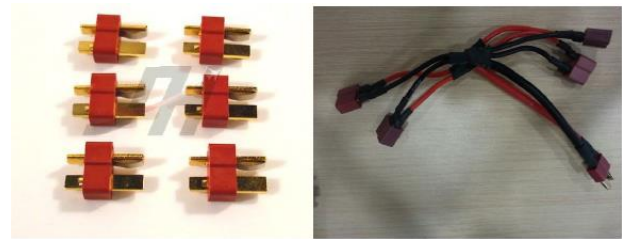


Fig. 6 Dean Plug male 4 Female to 1 male

4 to 1 Conversion Adapters

The 4 to 1 conversion adapters (Fig.6) are, 4 Dean Plug female to 1 Dean Plug male. The input for the female plug is the male plug soldered on the ESC wire. The output for the male plug of the adapter is to the battery. Only 1 adapter is required to power up the 4 ESC using 1 battery source.

Power Module

A power module has a 6-pos connector that provides +5.3V voltage and 2.25A current. It is needed as we need a connector to power the ArduPilot Mega. The power module accepts a maximum input voltage of 18V (up to 4S Lipo battery) and maximum current of 90Amps.



Fig. 7 Power Module

ArduPilot Mega 2.5

The ArduPilot Mega 2.5 is a complete open source autopilot system. The Ardupilot version that was used has a plug and play design, no assembly or programming is required. However there is a function to allow users to modify the programming codes using the Arduino software. The function of this devices is to turn any fixed, rotary wing or multirotor vehicle (even cars and boats) into a fully autonomous vehicle; capable of performing programmed GPS missions with waypoints.



Fig. 8 ArduPilot Mega 2.5

Fig. 9 AR 8000 DSMX 8 Channel

Receiver

The Receiver used is, 2.4GHz 8 channel. It comprises of a compact main receiver that is linked to an even smaller satellite receiver by a 6" lead, the AR 8000 acts as a pair of extremely sharp RF "eyes" that each take in a slightly different view of the signal environment. Information from both receivers is then processed by patent pending Spektrum software that takes information from each receiver and combines it to form the most vivid "picture" of an RF signal. Using 2.4GHz means that our receiver will not be interfered with other frequency unlike using crystals which could be interfered with when there are other flyers using the same frequency. After binding the receiver to the transmitter, it is forever bind to the particular transmitter which is why it will not interfere with others.

4. SIMULATION MODELS

Using the equations of motion derived earlier and the dimensions of the Quad-rotor constructed in previous section, a mathematical model is created and implemented on MATLAB Simulink. As Shown in Fig 17, the inputs for the quad-rotor simulation are the desired X-axis, Y-axis and Z-axis positions. The outputs are the linear and angular acceleration which will be integrated twice to obtain the respective positions and angles. Before designing and testing the controller, the model created was tested to ensure that its response are similar to the Quad-rotor dynamics.

4.1 Control Design

The controller must ensure that the plant's (Quad-rotor Model) output follows the reference input closely. A closed loop control structure is used in this project because it attenuated disturbances significantly and also compensate for any changes in the plant. Uneven weight distribution of Quadrotor can also be compensated by closed loop feedback.

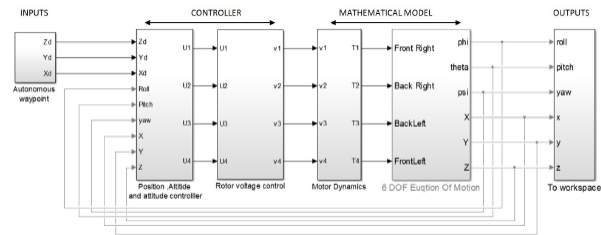


Fig. 10 Simulink Model of Quad-rotor

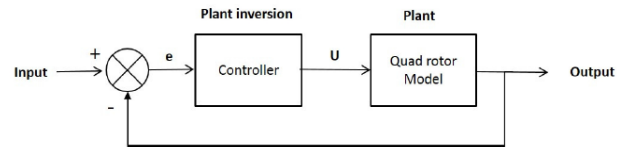


Fig. 11 Closed loop Control

The closed loop controller has to maintain output close to the input. Inherently closed loop control performs an inverse of the plant dynamics. Since the dynamics of the Quad-rotor is known, the controller gains can be tuned and an approximate plant inversion can be achieved. The resultant control signal 'U' is fed into the model to produce the desired output.

4.2 Controller Approach

Controller is constructed with PD loops. These PD loops inside the attitude, altitude and positional control generate the respective control signal.

The quad-rotor will use the Proportional Derivative control system (PD) to determine its response and settling time. The PD controller is a closed loop feedback system which will use the output control signal and feedback it to the initial input signal. The difference of both signals will be calculated and adjustment will be made.

In this case, the actual position and orientation of the quadrotor (output signal) will be feedback and being compared to the desired position and orientation (input signal) which will allow the system to adjust the process control input accordingly.

Control error is expressed as

$$e(t) = r(t) - y(t)$$

where $e(t)$ is the error signal. $r(t)$ is the input signal and $y(t)$ is the output signal.

Proportional Control

The proportional term depends only on the control error. It can control any stable plant. Increasing the proportional gain will decrease the response time of the control system. However, if the proportional gain is too high, oscillation will occur and system will become unstable. The proportional term is given by:

$$u(t) = K_p e(t)$$

where $u(t)$ is input signal to the plant. K_p is the proportional gain and $e(t)$ is the error signal.

Derivative Control

The derivative term gives a decreasing output if the process variable is increasing. It acts on the rate of change of the error. Increasing the derivative term will cause the system to react strongly to the changes in the error and therefore increase the speed of the system response. The derivative term is given by:

$$u(t) = K_d d[e(t)]/dt$$

where K_d is the derivative gain.

PD Tuning

In order to determine value of K_p and K_d , it is required to tune the controller until it reaches the desired response. The tuning of the proportional and Derivative values is achieved using Ziegler-Nichols Method and in most cases through manual tuning and trial and error. There is also a Simulink function to automatically tune the values, but further tuning is still required to obtain satisfactory response.

4.3 Control Architecture

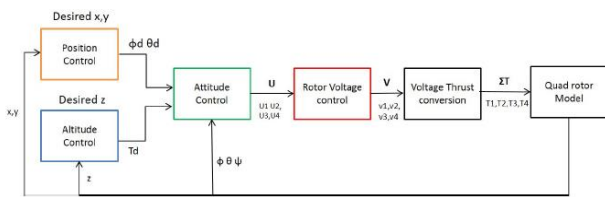


Fig. 12 Control Architecture

Roll, pitch and yaw are controlled by Inner PD loops, while the altitude and positions are controlled by the outer PD loop. The outer loop produces the desired values for the inner loop. Following section will focus into each control blocks in detail.

4.3.1 Attitude Control:

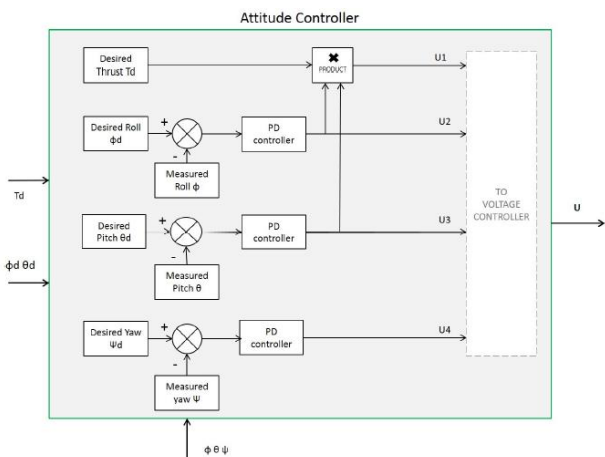


Fig. 13 Inner loop -Attitude Control

This inner loop is responsible for the control of Roll, Pitch and yaw. In this project, Quad-rotor is assumed to be symmetric about xy axes i.e. the pitch and roll controls are almost similar. Pitching down moves the Quad-rotor forward along X-axis while Rolling moves it side wards.

ROLL: The Control signal U2 is generated from the roll error fed through a PD control. This signal holds the roll attitude of the Quad-rotor.

PITCH: U3 is generated from the pitch error fed through a PD control. This signal holds the pitch attitude of the Quadrotor. As mentioned earlier the P&D values of roll and pitch are almost similar.

YAW: U4 is generated from the yaw error through a PD control. This signal holds the yaw attitude of the Quad-rotor. The measured $\phi\theta\psi$ are the actual feedback values from the output of the model.

4.3.2 Autopilot:

Altitude Control

This outer loop control generates the desired thrust T_d required to lift the Quad-rotor to the desired altitude. It is manipulated from the altitude error fed through a PD controller. T_d is then passed to the attitude controller for the lift vector compensation due to roll and pitch. Hence the control signal U1 will hold the desired altitude of the Quad-rotor even during roll and pitch. The following block shows the control loop and the respective signals inside the altitude controller. The feedback provides the measured altitude of the Quad-rotor.

Altitude control

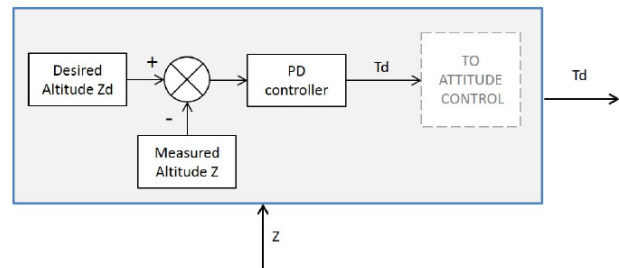


Fig. 14 Outer Loop -Altitude Control

Position control:

This outer loop generates the desired roll and pitch angles required to move the Quad-rotor to the desired X and Y positions. Feedback data from the model provides the measured X and Y. The position error signal after summing junction is fed through a PD control and it generates the desired roll and pitch angle. As mentioned earlier, its output is then fed to the attitude control to generate the control signals U2 and U3.

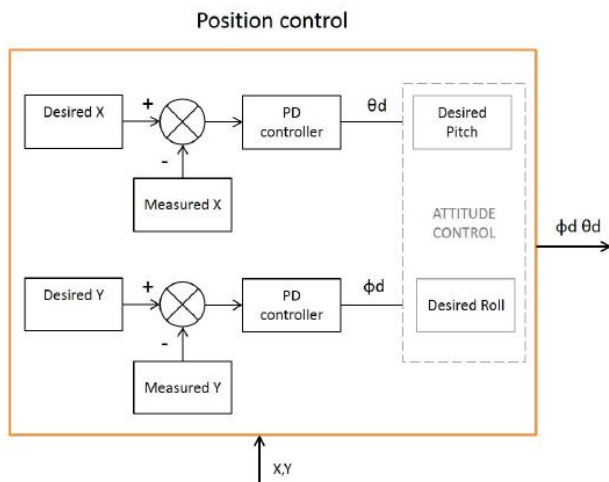


Fig. 15 Outer loop - Position control (X, Y)

4.3.3 Rotor Voltage Control:

The 4 control inputs generated by the inner and outer loops cannot be fed directly to the Quad-rotor Model because its inputs are in terms of Thrust for the 4 individual rotors. The control signals 'U' are combined to produce voltage signal for each rotor. These combinations are based on the dynamics of the Quad-rotor to exhibit the respective Flight parameter. The combinations was mentioned earlier under the Motor dynamics and also shown below in the block diagram. The output voltage 'V' is then converted to thrust by the modelled motor.

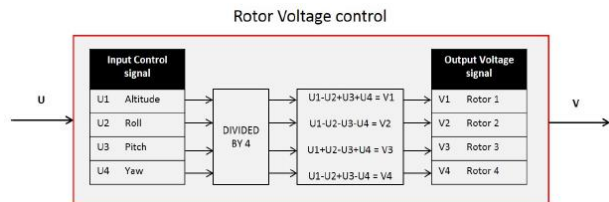


Fig. 16 Rotor voltage control

5. SIMULATIONS AND FLIGHT TEST RESULTS

The Flight test data from the Lab was captured using the Opti-track cameras and was imported into MATLAB for comparison with the simulated results. Technical difficulties and observations from comparison are also discussed.

5.1 Simulink Test Results:

Attitude control:

A pulse signal was fed as the desired Roll, pitch and yaw inputs to the designed controller for testing. The plots on the left segment in Fig 17 shows the input to the controller while the right segment shows the respective response from the model

simulated. The response plots clearly depict the attitude control is working as desired.

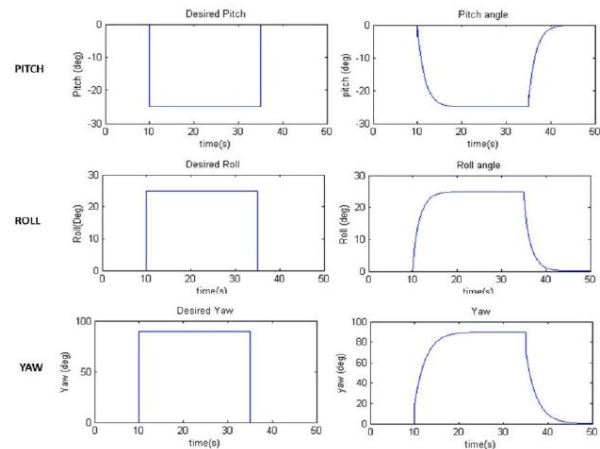


Fig. 17 Attitude control Simulation plots

Position and Altitude Control:

A step input with a magnitude of 10 was fed to position and altitude controller. The desired 10 m input is expected to move the Quad-rotor model to 10 m position and hold. As mentioned earlier the position controller's output will command the desired roll and pitch angle to the attitude control. The right section of the plots in Fig 18 shows how the roll and pitch angles change in order to reach the desired Y and X position respectively.

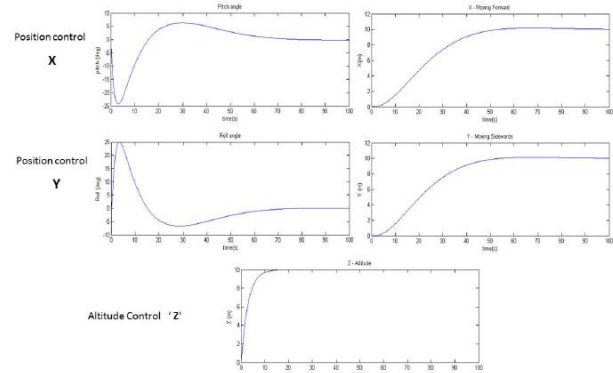


Fig. 18 Position control simulation plots

Waypoint Navigation:

An autonomous waypoint flight was demonstrated in Simulink using the model. A set of waypoints was fed to the position controller using timer control and the respective x, y, z position data was obtained from the simulation. The waypoints and the flight path of the modelled Quad-rotor in 3D are shown below. The plots below shows that the positional controller performs as desired.

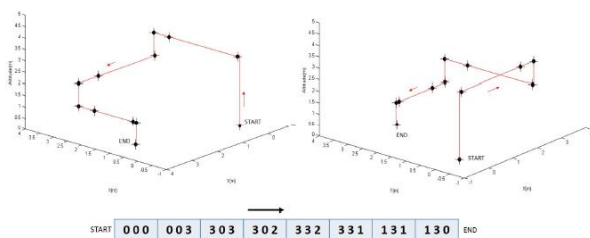


Fig. 19 3D Flight path -Waypoint Navigation using position control

5.2 Actual Flight Test

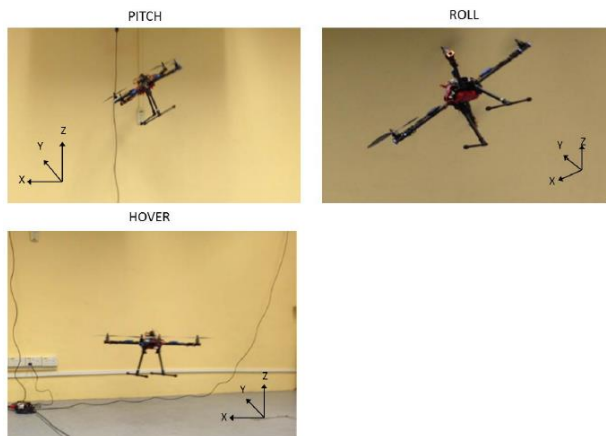


Fig. 20 Flight Test Pictures

The constructed Quad-rotor was put to flight with the Ardupilot microcontroller. The quad-rotor's movement, the positions of the quad-rotor (X, Y and Z-axis) was captured by the Opti-Track system during flight tests. The following pictures were taken during one of the actual tests. They show the pitch, roll and take off of the quad-rotor respectively during the test flight.

Technical challenges

The Quad-rotor was unstable during most of its flights. Initially the uneven weight distribution was thought to be the reason for instability. Adding counterweights, to spread the weight distribution equally, failed to solve the issue. Even trimmed flight could not completely cease the undesired yawing and rolling motion. After repeated test flights it was discovered that the motors used did not produce similar thrust though their specifications are similar.

All 4 rotors were tested using a thrust meter attached to the same ESC. The readings were taken for different throttle settings in the radio controller. The following table and the graph clearly depicts that the motors were not generating similar thrust for the same radio input. There were differences of about 100g of thrust for certain throttle settings.

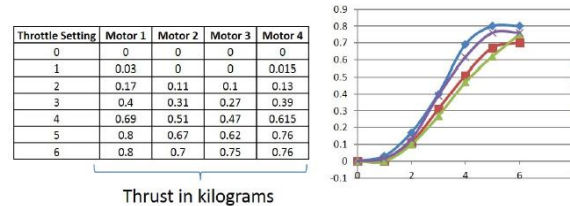


Fig. 21 Motor Thrust Indifferences graph and Table

Simulation vs Test Flight

Besides unstable flight, certain segments of the flight test was captured well by the Opti-Track system. It was then imported into MATLAB. In Fig 22 below the plots on the right segment was captured from 3 different test flights while the plots on left segment was obtained from the Simulation performing similar outcomes of the captured data.

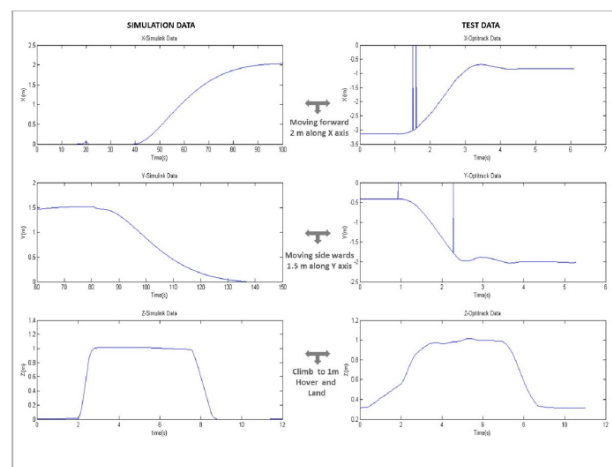
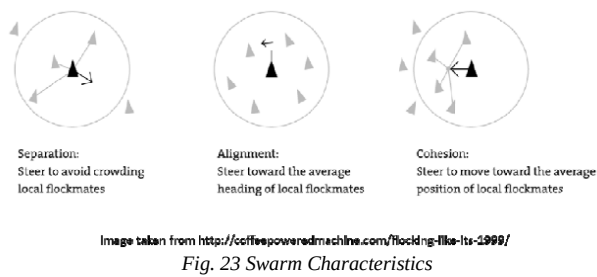


Fig. 22 Simulation v Flight Test data

From comparison the envelope of both the responses are similar with simulated response being more ideal. But the Response time of the simulation is much slower as compare to the test data. Reason being, the inputs for the actual flight test are the roll, pitch and yaw commands from the radio controller. No autopilots were used. Hence the response was fast. However, the simulation used the position controller which took longer time to compute the desired roll and pitch angles. Also the mathematical model of the system 100% similar as the actual quad-rotor and did not account for any external disturbances. These are the factors that caused the difference.

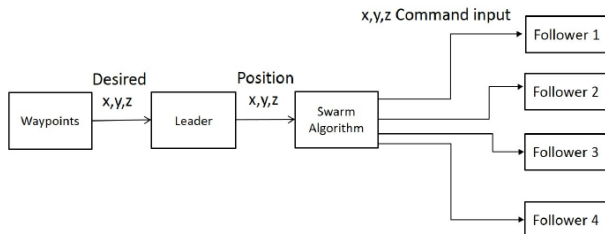
6. SWARM BEHAVIOR

Quad rotors can work in teams with the help of swarm algorithms. These algorithms are nothing but a set of rules that exhibit the three main characteristics of a swarm behaviour such as Cohesion, Separation and Alignment. The following Fig 23 provides more information about each characteristic.



6.1 Leader Follower Method

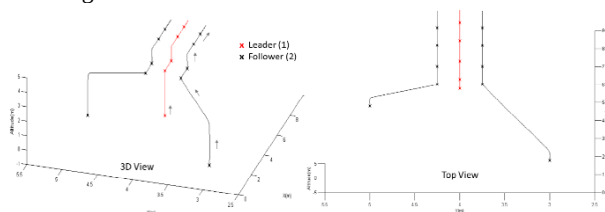
As the name describes, the leader leads a team of quadrotors. It acts as the reference point for the swarm. The actual positional information of the leader is fed into the swarm algorithm to generate the position command for the followers. The swarm algorithm makes sure that the followers are separated by a certain distance from the leader while following. The command input for each follower is computed depending on the position occupied by that follower in the formation.



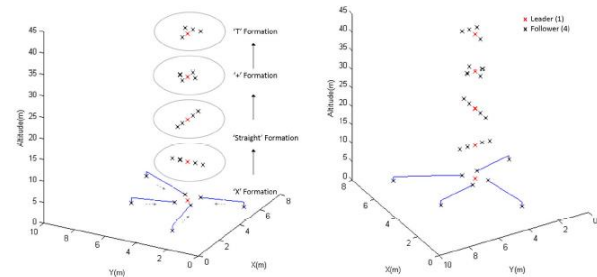
6.2 Swarm Simulation

The position control forms the basis for swarm implementation as the output of the swarm control is in terms of positions. The capability of the position control was displayed earlier (waypoint navigation for a single quad rotor). In the following simulation, the leader is provided with the waypoints to take off and move forward. The plots in Fig 25 shows a leader and 2 followers executing the following sequence of actions.

1. Quad-rotors Take-off from their initial position.
2. The followers identify the leader and move closer to it.
3. The 2 Followers travel behind the leader as it moves forward, maintaining the safety distance predefined as separation vector in the algorithm.



The formation pattern can be changed by changing the separation vector. In the following simulation shown in fig 26, a leader and 4 Quad-rotors demonstrate different formations as they climb upwards following their leader. From their initial position, they move closer to the leader. Leader varies his altitude only. The followers change their positions due to the varying separation vector.



7. CONCLUSION

This paper presented a Quad-rotor model constructed in our UGS lab and a model simulated in MATLAB. Though the simulation managed to achieve a stable and autonomous flight, the actual Quad-rotor exhibit moderate unstable controlled flight due to technical challenges mentioned earlier. Several remedial measures such as introducing the offsets to the motor to linearize the motor output failed to solve the issues.

Working with Motion capture system was completely new and knowledgeable. The ability to collect variety of flight performance data from the system without the need of additional sensors aboard the platform is greatly appreciated. From the results of both simulation and actual flight test, the students learned the limitations of simulation. This learning will improve the way we design and make our future designs more accurate and realistic.

REFERENCES

- [1] Bouabdallah, S. 2007. *Design and control of Quadrotors with Application to Autonomous Flying*. Université Aboubekr Belkaid.
- [2] Christiansen, R. S. 2004. *Design Of an Autopilot For Small Aerial Vehicles*. Brigham Young University.
- [3] M. Fadhil Abas, S. A. n.d. *Leader-Follower Formation Control of Multiple Quad-rotor Aerial Vehicles*. Chiba, Japan: Chiba University.
- [4] Soon Hooi Chiew, T. H. Go, 2012. *Swarm Coordination with Multiple Quadrotors*. Master thesis, Nanyang Technological University, Singapore.
- [5] Robert Stengel. 2012 *Aircraft Flight Dynamics*. [Online].
- [6] <http://www.princeton.edu/~stengel/MAE331Lecture8.pdf>
- [7] Andrew Gibiansky, "Quadcopter Dynamics, Simulation, and Control,"
- [8] Ardupilot.com. n.d. APM Open source autopilot. [online] Available at: <http://ardupilot.com/>
- [9] Copter.ardupilot.com. n.d. APM 2.5 and 2.6 Overview. [online] Available at: <http://copter.ardupilot.com/wiki/common-apm25-and-26-overview/>

- [10] Douglas, S. n.d. APC Propellers, props for model airplanes. [online] Available at: <http://www.apcprop.com/v/index.html>
- [11] Foxtechfpv.com. 2014. TMOTOR MT2212 KV980. [online] Available at: <http://www.foxtechfpv.com/tmotor-mt2212-kv980-p-656.html>
- [12] Hefny, M. 2012. Technical Adventures: QuadCopter Stabilization & Control System "X & Plus Configuration". [online] Available at: <http://technicaladventure.blogspot.sg/2012/09/quadcopter-stabilizationcontrol-system.html>
- [13] Icharger.co.nz. 2011. Introduction To Lipo Batteries. [online] Available at: <http://www.icharger.co.nz/articles/ArticleId/6/Introduction-To-Lipo-Batteries.aspx>
- [14] Platinum Label LipoDocier. n.d. [e-book] Revolectrix. p. 5. http://revolectrix.com/support_docs/item_1406.pdf
- [15] Code.google.com. 2012. APM2RC - arducopter. [online] Available at: <https://code.google.com/p/arducopter/wiki/APM2RC>
- [16] Aerospace.illinois.edu. n.d. UIUC Propeller Data Site. [online] Available at: <http://aerospace.illinois.edu/m-selig/props/propDB.html>
- [17] Beikrit, S., Falaschini, C., Abdolhosseini, M. and Capotescu, F. n.d. QballQuadrotor Helicopter. [report].
- [18] C.Cody, B. 2012. *Simulation and Position Control of a Quadrotor UAV*. [report].
- [19] Angeletti, G., Valente, J. R. P., Locchi, L. and Nardi, D. n.d. *Autonomous Indoor Hovering with a Quadrotor*. [report].
- [20] Cavalcante Sá, R., C. De Araújo, A. L. and T. Varela, A. 2013. *Construction and PID Control for Stability of an Unmanned Aerial Vehicle of the Type Quadrotor*. [report].
- [21] Cutler, M. J. 2010. *Design and Control of an Autonomous Variable-Pitch Quadrotor Helicopter*. [report].
- [22] Hussein, W. M., El-Khatib, M. M., Elruby, A. Y. and Haleem, H. M. n.d. *Quad Rotor Design, Simulation and Implementation*. [report].
- [23] Kim, J., Kang, M. and Park, S. 2009. *Accurate Modeling and Robust Hovering Control for a Quad-rotor VTOL Aircraft*. [report]. 43
- [24] Orsag, M. and Bogdan, S. n.d. *Influence of Forward and Descent Flight on Quadrotor Dynamics*. [report].
- [25] Palm, J., Nelson, A. and Bradford, A. 2010. *Quad-Rotor UAV project*. [report].
- [26] Raza, S. A. and Gueaieb, W. n.d. *Intelligent Flight Control of an Autonomous Quadrotor*. [report].
- [27] Roberts, J. F., Stirling, T. S., Zufferey, J. and Floreano, D. n.d. *Quadrotor Using Minimal Sensing For Autonomous Indoor Flight*. [report].
- [28] Tobias, P., Krogstad, T. R. and Gravdahl, J. T. n.d. *UAV Formation Flight using 3D Potential Field*. [report].
- [29] Elkaim, G. H. and Kelbley, R. J. n.d. *A Lightweight Formation Control Methodology for a Swarm of Non-Holonomic Vehicles*. [report].
- [30] Flyingmachinearena.org. n.d. *Flying Machine Arena*. [online] Available at: <http://flyingmachinearena.org/>
- [31] Kushleyev, A., Mellinger, D. and Kumar, V. n.d. *Towards A Swarm of Agile Micro Quadrotors*. [report].
- [32] Yasir Amir Khan Niazi. 2010. *Modeling and neural control of quadrotor helicopter*. Berlin: LAP Lambert Academic Pub.
- [33] Putro, I., 2011. *Modeling and control simulation for autonomous quadrotor*. 1st ed. Saarbrücken: LAP Lambert Academic Pub

Using the Paparazzi UAV System for Scientific Research

Gautier Hattenberger*, Murat Bronz† and Michel Gorraz
ENAC, MAIAA, University of Toulouse, F-31400 Toulouse, France

ABSTRACT

This paper presents an overview of the *Paparazzi* UAV system and its recent use in scientific research. *Paparazzi* is an open-source project that aims at providing a complete solution to fly fixedwing aircraft and rotorcrafts. Several hardware boards and sensors are also developed within the project. Since several years, it has been used by various institutes for scientific research. The recent use on scientific research for meteorological studies is presented as an example.

1 INTRODUCTION

For the last ten years many open-source autopilot systems have been developed by various communities [1]. Some of them are designed for hobby flights while others aim at more professional applications. The *Paparazzi*¹ project [2], started in 2003 by teachers of the French Civil Aviation University (ENAC), is one of the oldest with the goal to provide an affordable and reliable platform for education and academic research. It is now supported by other institutes such as MAVLAB of the TU-Delft, individual developers and some private UAV companies from several countries.

This paper proposes an overview of the *Paparazzi* system with its ground and airborne software architecture, and some of the recent hardware solutions developed for the project. As an example application, a meteorological study led by ENAC and Météo-France is presented.

2 OVERVIEW OF THE PAPARAZZI SYSTEM

The global view of the system (see figure 1) is very typical of UAV systems and has not much changed since the beginning of the project:

Ground segment: all the ground software and hardware infrastructures that are used to prepare, monitor and analyze the flight.

Airborne segment: the aircraft, its hardware parts including payload and all the embedded software to control the flight (from stabilization to decision making).

Communication segment: defines all the communication links and protocols between the ground and airborne segments.

*gautier.hattenberger@enac.fr

†murat.bronz@enac.fr

¹<http://paparazziuav.org>

Safety link: safety remote control that can be separated from the ground station for short range direct control.

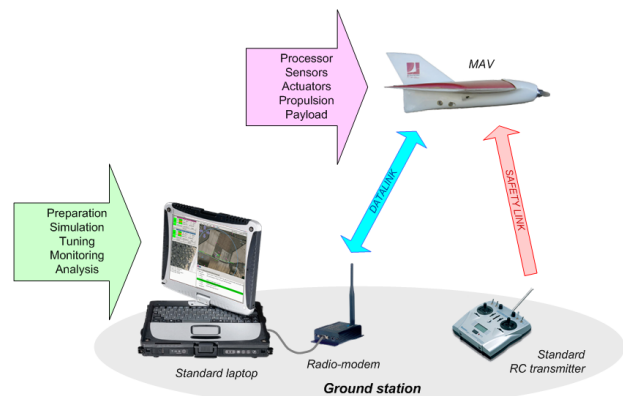


Figure 1: Global view of the *Paparazzi* system

3 FLYING PLATFORMS

The *Paparazzi* system initially designed for robust small fixed-wing aircrafts in 2003, however during the following years several other configurations and concepts have been integrated to the system such as high-aspect ratio gliders, multi-rotors, and transitioning vehicles [3, 4]. Today, *Paparazzi* flies on 16cm *Quark* [5] up to 4.3 meter spanned Adler UAV from University of Stuttgart [6] as shown in figure 2. The popular UAV platform *AR.Drone 2* from Parrot² (figure 3) can also be used to run the *Paparazzi* software [7].



Figure 2: *Quark* and *Adler* UAV as two examples of *Paparazzi* flying fixed-wing aircrafts

4 SOFTWARE ARCHITECTURE

The software architecture (see figure 4) is organized on the ground side around several agents connected through the *Ivy*

²<http://ardrone2.parrot.com/>



Figure 3: *AR.Drone 2* with additional GPS and *Asctec Hummingbird* frame as two examples of *Paparazzi* flying rotorcrafts

software bus [8].

One or several UAVs (in blue) are connected to the ground (in brown) with RF-modems or WIFI connections. The *link* agent manages the ground-based radio modem and distributes telemetry data across the network (a single computer, a local network or the internet) where it can be used locally or remotely by the *server*, an agent that logs, distributes, and pre-processes these messages. A set of synthetic messages are then dispatched to other agents, such as the ground control station, or *gcs*, that provides a graphical user interface. Some other tools can be used with the *Ivy* bus such as the *messages* agent that display all the messages, or simulator agents (aircraft, environment) that allow to test the flight plans in various conditions for instance.

Usually, the ground softwares are running on a standard laptop. The project mainly support the Debian/Ubuntu GNU/Linux operating system, but other Linux based distributions and MacOSX systems are possible. In addition, the ground station can run on low cost, low power boards such as the popular RaspberryPi³.

4.1 Ground station

The standard ground station interface is one of the oldest part of the system and as not much change during the last years, although some improvements have been done like a vertical view. For historical reasons, the *gcs*, like most of the ground agents, is coded in OCaml and is using GTK as a graphical toolkit.

The *gcs* propose the following features:

- simultaneous flying multi UAV support
- 2D Map capable of displaying Google Satellite, OpenStreetMaps Images and Microsoft Satellite Maps
- real-time movable waypoints
- system status rapid overview with the strip (see figure 6)
- supports rotary and fixed-wing e.g. Airplanes, helicopters, coaxial and quadrotors

³<http://www.raspberrypi.org/>

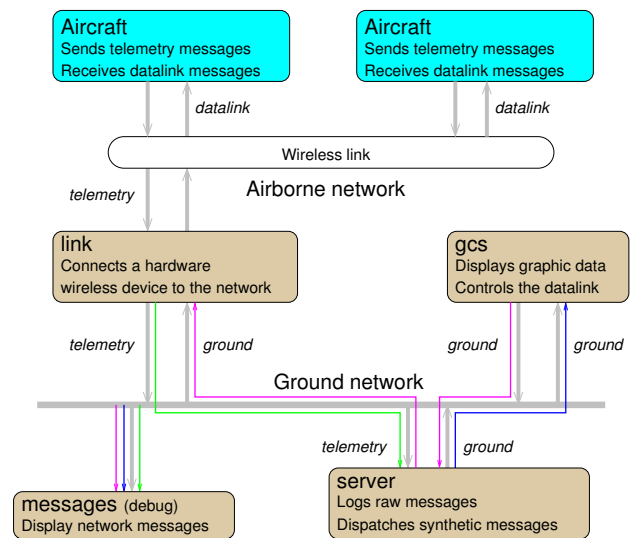


Figure 4: Global view of the *Paparazzi* system

- definable Hotkeys for quick simple in the field control
- voice status output
- full freely configurable GUI layout with various plugins (vertical view, video feedback, text and graphical widgets integrated to the 2D map)

Recently, an Android application (see figure 7) has been developed to be used as a remote *gcs*, with the Android device connected to the main ground station and displaying aircraft position, status and flight plan. Basic high level controls are also possible, and more features will be added in the future to be a viable alternative to the standard interface.

4.2 Airborne code

Based on the old airborne code design, a great effort has been made in order to have a clear separation of the different layers (see figure 8) that would ease the transition to a real-time OS based solution (the current one still use a static scheduling). Especially the data between the sensors and the estimation filters are managed by a publisher/subscriber mechanism inspired by the ground station *Ivy* bus. The resulting estimated state is then published in a blackboard type interface for the control and navigation loops.

Previously, the attitude estimation was based on infrared thermopiles, but is now replaced by IMU-based filters such as complementary filters or kalman filters. Some advance filtering solutions have been tested using the *Paparazzi* system [9].

4.3 Flight plans and mission control

The *Paparazzi* system is using a unique flight plan language, based on *XML*, that allows to:

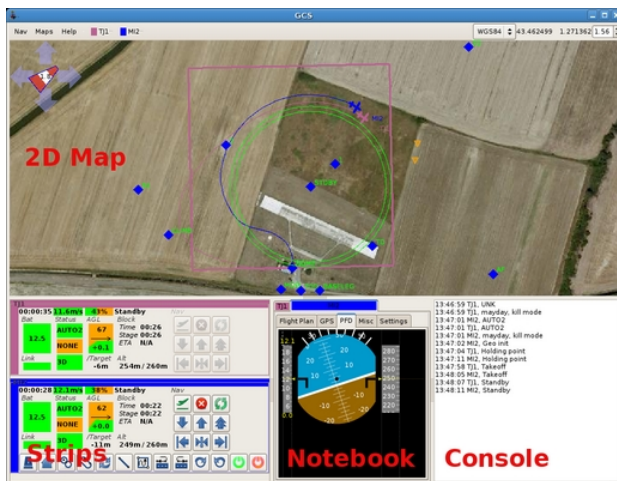


Figure 5: The ground station interface

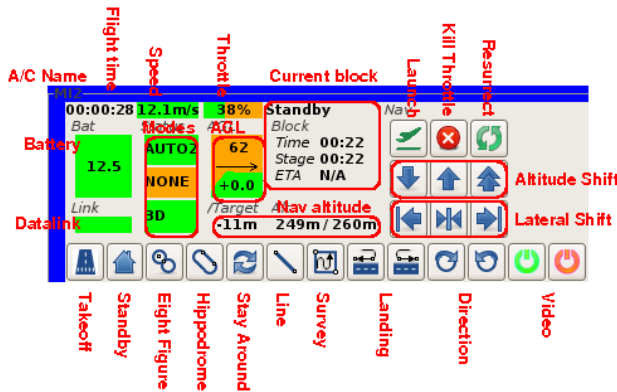


Figure 6: The GCS strip allows a rapid overview of the UAV status

- define a list of waypoints
- define sequences of small tasks (called *blocks*)
- perform *for* and *while* loops
- perform actions based on global or local exceptions
- call a list of predefined navigation patterns (circle, line, survey,...)
- call any custom navigation function written in C

The flight plan description is used to generate a C code compiled with the rest of the autopilot. This static behavior allows to use custom and complex navigation patterns (for instance in photogrammetry applications). The main drawbacks are that since the list of waypoints and blocks is defined before



Figure 7: The PPRZonDroid application running on an Android tablet

the mission, it is not possible to add new ones at runtime (but the operator can move the waypoints and switch to another block).

In order to overcome these limitations in the case of dynamically generated missions (especially when using a task planning system), a dynamic mission controller has been added. The principle is that only basic tasks (goto, circle, segment) can be used, but they can be loaded in queue list where they will be executed in sequence. A new task can be added at any time (at the end, before existing tasks, or replacing all current tasks) and be received from the datalink system, from an internal module, or from an extra embedded computer.

4.4 Communications

The communication's protocol and the standard set of message used by *Paparazzi* are now reaching their limits. Especially the number of available slots for downlink messages is rather small, while many of this messages could be reorganized in a better way. Some work have been started in this direction. One of the future improvements should also come from an evolution of the current encapsulation protocol shown figure 9 by introducing classes of message or component ID.

An other plan for the future is to support the *Mavlink* protocol⁴ or at least a part of it.

4.5 Real-Time OS

Since the beginning of *Paparazzi*, the scheduling of the autopilot tasks is done statically. On the old generation of boards, the computation power exclude the use of too complex real-time operating system (RTOS). The new generation of micro-controllers allows to use such systems. Several Open-Source options are available such as *NuttX* or *FreeRTOS*.

In order to develop embedded logging capabilities, a

⁴<http://qgroundcontrol.org/mavlink/start>

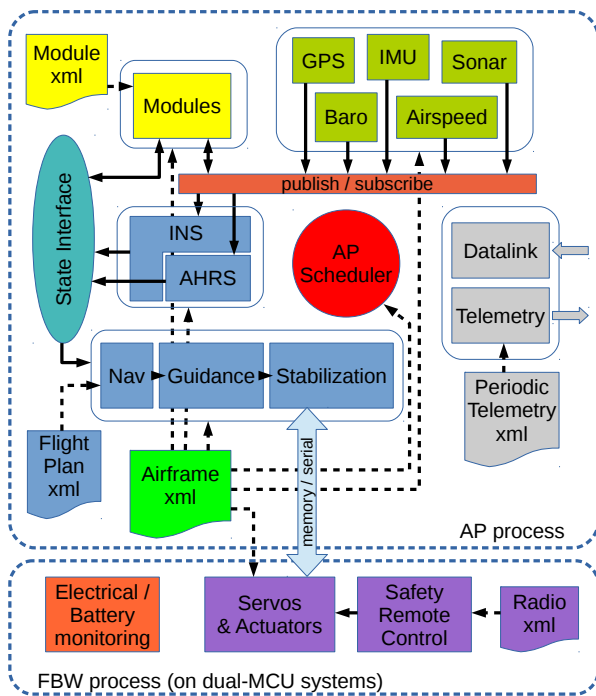


Figure 8: The fixedwing airborne software architecture

RTOS was needed in order to guarantee that the autopilot operations are not delayed when writing on the SD-card. The choice as been made to use *ChibiOS*⁵, since it is pretty powerful OS that offers all the necessary components needed for logging.

Currently, the RTOS handles only two tasks, one for logging and one for the rest of the autopilot. For the future developments of *Paparazzi*, it is planned to split the different tasks of the autopilot, allowing accurate timing of the critical tasks (state estimation, control) while others have low priorities (navigation, payload control, logging,...).

On the *AR.Drone 2* platform, the autopilot runs on a linux based OS. It is then possible to easily use multi-threaded processes, and run vision algorithms using the integrated cameras for instance.

5 HARDWARE SOLUTIONS

Many controller and sensor boards have been developed for the *Paparazzi* project. The most recent autopilot boards are based on the STM32 micro-controller line from ST-Microelectronics.

The latest autopilot board developed by ENAC is called *Apogee* (see figure 10). It has the same size as the previous generation, with a micro-SD card slot and more powerful micro-controller unit capable of handling the autonomous

⁵<http://www.chibios.org>

Figure 9: The current standard encapsulation protocol of *Paparazzi* messages used for RF binary communications

navigation of the MAV, a wider range of payload with more inputs/outputs, and high speed data logging (see characteristics in table 1). The design has been done especially thinking of scientific applications requiring data logging for not losing important data over the telemetry link and without the need of an external board.

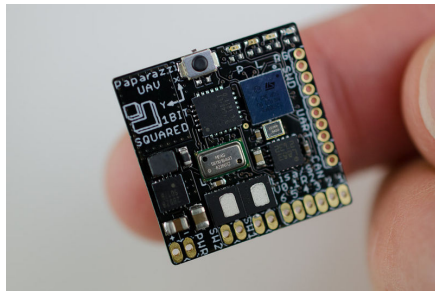
Figure 10: *Apogee* autopilot controller board

Other autopilots have been developed for the *Paparazzi* project with different objectives, such as easier integration on rotorcrafts (*KroozSD* board) or the smallest autopilot *Lisa/S*⁶ (see figure 11).

All of these boards are integrating IMU sensors (usually from the MPU line from Invensense), barometer and even GPS U-blox module for the *Lisa-S*.

⁶<http://1bitsquared.com/>

Processor	STM32F405
IMU	MPU-6050
Barometer	MPL3115A2
Logging	MicroSD slot over SDIO interface
Connectivity	6 PWM, 3 UART, 2 I2C 1 SPI, 1 SWD, 1 USB 4 AUX (PWM, ADC, GPIO)
Remote control	1 PPM input, 1 serial input
Power	1.5A switching power supply 1 5V/500mA power switch
Size x weight	53x25 mm, 10.4 grams, 4 layers PCB

Table 1: *Apogee* autopilot characteristicsFigure 11: *Lisa/S*, the smallest complete autopilot

6 A SCIENTIFIC APPLICATION: METEOROLOGICAL STUDIES

The *Paparazzi* system serves as a complete package for the scientific research missions. Its highly flexible architecture and open-source software helps the researchers and the scientists for their studies. As an example, the *VOLTIGE* (Vecteur d'Observation de la Troposphère pour l'Investigation et la Gestion de l'Environnement) project is presented here.

6.1 Objectives

The main objective of the *VOLTIGE* project is to fly with a fleet of UAVs simultaneously in and out of fog and get temperature, pressure, humidity, solar radiation, cloud and fog detection, and turbulence measurements. The UAVs that are used on the mission are lighter than 2 kg in order to facilitate flight permission and regulation issues. This is a limiting factor for the payload capacity of each vehicle. Different sensors are integrated into different UAVs according to the required flight type. *VOLTIGE* project has mainly three type of flight profiles, turbulence measurements are taken from steady horizontal flight inside the fog at different levels, radiation measurements are taken above the fog which implies the detection of the fog boundary and vertical profile measurements are taken by flying from ground level to the upper boundary of the fog.

Paparazzi System has an automatic collision avoidance system which serves an important role on the low visibility missions such as flying inside the fog with multi-UAVs.

6.2 Platforms Used

Initially, two different platforms are used for the *VOLTIGE* flights which are shown in figure 12. *Funjet* is more suitable for vertical profile measurements as it can climb more efficiently to higher altitudes. *EasyStar* is mostly used for the horizontal flights at constant altitude as it is more stable and has more payload capacity.

Figure 12: The two type of platforms used in *VOLTIGE* project

Several flight campaigns have been performed with these two airframes, and concluded that the *Funjet* airframe is reaching its maximum take-off weight limits with the additional equipments used, which makes it harder to hand-launch and land in small areas. On the other hand *EasyStar* airframe serves well for carrying the required payload with an easy hand-launch and landing characteristics but lack of climbing to the required altitude. As the goal of the project is to fly several UAVs at the same time, both airframes suffer from a short endurance performance (which is limited to 35 minutes) for the complete mission.

Figure 13: *FENIX* UAV designed for *VOLTIGE* project mission requirement

In order to overcome these problems a new airframe is designed particularly for the *VOLTIGE* project mission requirements. A flying-wing configuration, called *FENIX*, with four control-surface is designed to separate elevator and aileron controls. The fuselage is mainly optimized in order to facilitate the integration and operation of the selected payload sensors and electronic cards. The endurance performance has

been increased up to two hours for the level flight and kept over one hour for the climbing flight for the vertical profiles of the mission. This improvement will mainly help for the safety of the multi-UAV operation.

7 CONCLUSION AND FUTURE WORK

The current state of the *Paparazzi* system has been described in the paper. Latest hardware and software capabilities are explained on a recent scientific application which requires robust operation in harsh weather conditions with multiple UAVs flying simultaneously while recording synchronized data for meteorological measurements.

Many developments are still undergoing to make the *Paparazzi* autopilot safer for real world operations or academic research, easier to use for new users while keeping its versatility. The efforts should be focused on the proper support of a real-time operating system, an improved ground control station and a need definition of the communication layer and its messages.

ACKNOWLEDGMENTS

The authors would like to thanks all the *Paparazzi* community for the development, testing and support to this project.

REFERENCES

- [1] Daewon Lee Hyon Lim, Jaemann Park and H.J. Kim. Build your own quadrotor. *IEEE Robotics & Automation Magazine*, September.
- [2] Pascal Brisset, Antoine Drouin, Michel Gorraz, Pierre-Selim Huard, and Jeremy Tyler. The *Paparazzi* solution. *MAV2006, Sandestin, Florida*, 2006.
- [3] Pranay Sinha, Piotr Esden-Tempski, Christopher A Forrette, Jeffrey K Gibboney, and Gregory M Horn. Versatile, modular, extensible vtol aerial platform with autonomous flight mode transitions. In *Aerospace Conference, 2012 IEEE*, pages 1–17. IEEE, 2012.
- [4] C De Wagter, D Dokter, G de Croon, and B Remes. Multi-lifting-device uav autonomous flight at any transition percentage, 2013.
- [5] Murat Bronz, Jean-Philippe Condomines, Gautier Hattenberger, et al. Development of an 18cm micro air vehicle: Quark. In *Proceedings of the International Micro Air Vehicle Conference and Flight Competition IMAV 2013*, 2013.
- [6] Marc Schwarzbach, Uwe Putze, Ursula Kirchgaessner, and Maria v Schoenermark. Acquisition of high quality remote sensing data using a uav controlled by an open source autopilot. In *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 595–601. American Society of Mechanical Engineers, 2009.
- [7] Bart Remes, Dino Hensen, Freek van Tienen, Christophe De Wagter, Erik van der Horst, and Guido de Croon. *Paparazzi: how to make a swarm of parrot ar drones fly autonomously based on gps*. In *International Micro Air Vehicle Conference and Competition IMAV2013*, September 2013.
- [8] Marcellin Buisson, Alexandre Bustico, Stéphane Chatty, Francois-Régis Colin, Yannick Jestin, Sébastien Maury, Christophe Mertz, and Philippe Truillet. Ivy: un bus logiciel au service du développement de prototypes de systèmes interactifs. In *Proceedings of the 14th French-speaking conference on Human-computer interaction (Conférence Francophone sur l'Interaction Homme-Machine)*, pages 223–226. ACM, 2002.
- [9] Jean-Philippe Condomines, Cédric Seren, Gautier Hattenberger, et al. Nonlinear state estimation using an invariant unscented kalman filter. *AIAA Guidance Navigation and Control Conference*, pages 1–15, 2013.

Autonomous Aerial Mapping and Observation Task for IMAV2014 Competition

Emmanuel Colles¹, Fabien Nollet¹, Bertrand Vandeportaele^{*2,3}, and Gautier Hattenberger^{†1}

¹ENAC, MAIAA, University of Toulouse, Toulouse, France

²CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France

³Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France

ABSTRACT

The IMAV2014 Competition gives an important place to computer vision and aerial mapping. This paper presents the different strategies and algorithms that have been developed in order to tackle these particular problems. The first task consists in automatically reading the digit displayed by a seven segment panel on the front wall of a building. Robust algorithms for segments detection are used in order to extract the features from the scene. The second task requires to produce an aerial map of the competition place. In order to generate a coherent composite image, photogrammetry methods are used, including interest points pairing and bundle adjustment. We propose to mix 3D mapping methods and standard 2D image mosaicking in order to achieve the processing in the time available during the competition while obtaining an geometrically consistent map.

1 INTRODUCTION

The IMAV competitions are always pushing the limits in terms of micro air vehicle performances and autonomy. The 2014 edition gives an important place to real-world operation by setting the mission in a real military training village. Since many tasks require the use of computer vision, the reliability and the robustness of the algorithms are a key point. This article presents the strategies and the algorithms developed to tackle two particular tasks: detecting and reading a seven segment digit placed on a building front wall; and making an aerial mapping of the competition area.

For the first task, presented in section 2, several methods are available to extract a digit and recognize it. The selected solutions are using the knowledge on the digit color and shape, with a robust segment detection[1] and the use of an optical character recognition. The extracted pattern is then compared to a database to find out the recognized digit.

The aerial mapping task, developed section 3, is based on photogrammetry methods[2] and use some of the MICMAC

tools[3] and a new mapping framework that have been developed specifically for the occasion.

2 DIGIT RECOGNITION

Optical characters recognition is a problem that is studying in many fields as automotive to assist drivers by scanning road signs[4], security to recognize license plates[5], verify signature on cheque or human assisting as well to read zip code and make easier letters sorting. For the IMAV2014 Competition, one of the mission elements consists in observing the longest sequence of digits on a panel placed on the front wall of a building. The digit are going to change with a rate of thirty seconds. So there will be 60 different digits during the thirty minutes of the IMAV Competition.

At this point, it is not sure if the UAV will be landed or hovering in stable flight to manage this mission. With this constraint in mind, different solutions are developed to insure the best digit recognition depending of the UAV position and stability landed or not. In this perspective, section 2.2 present a process providing efficient results if the UAV is landed and, in section 2.3, a second process that should be used if the UAV is flying but stable.

2.1 Preprocessing

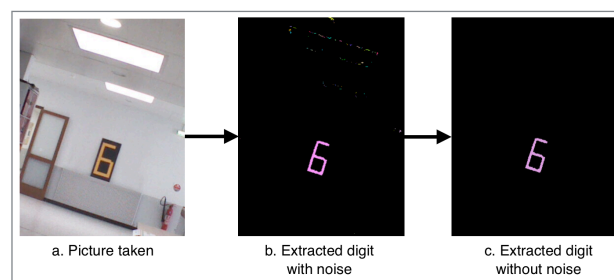


Figure 1: Pre-processing to extract digit bars

Although the end of the process is different, pre-processing is shared by both methods. This pre-processing exploits digit color and geometry of the pattern to read. By the use of the color, digit's bars are extracted from the image. In order to improve robustness, the HSL¹ color space

*bertrand.vandeportaele@laas.fr

†gautier.hattenberger@enac.fr

¹Hue Saturation Lightness

is preferred to RGB² representation. This allows extraction of pixels of a certain hue, quite independently of lightning conditions, which is especially useful for outside missions. Further to this extraction, some noise can interfere and a process that exclude isolated pixels or small sets of pixels based on labellisation is used. As shown on figure 1, digit's bars are correctly extracted, and the next step, digit recognition, can be carried out according to the current UAV state.

2.2 UAV landed

In this section, the UAV is considered landed and its camera is observing the whole digit panel even if the digit is on a picture corner. Due to the condition that the UAV is not moving, the digit panel should project in the same position in the different images grabbed from the camera. In this case, the processing is done on an integrated picture. This process allows to sum pictures and highlight differences in a sequence. If each bar has been turned on one time at least as shown figure 2, the complete 8 digit can be rebuild. When the 8 digit is rebuilt, the four corners are extracted by computing intersections between lines detected robustly using the RANSAC³ algorithm[6], independently of the panel orientation.

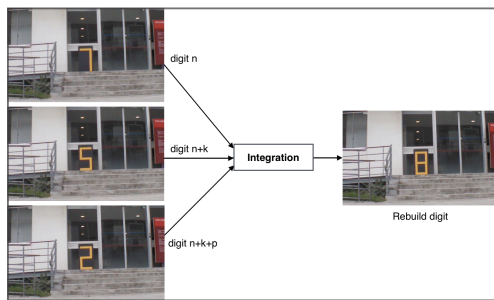


Figure 2: Integration of pictures to rebuild an 8 digit (all segments turned on)

Further to the extraction of the four corners, a bounding box is determined and a perspective transformation (homography) is applied to reshape the extract digit as shown figure 3 such that it fills an image of known size. This image is then compared to templates for recognition. Thanks to the homography, the process of decoding the digit is simplified, as it just consists in comparing the pixels values from the rectified image and an bank of ten reference images.

After this first extraction, it is not required to repeat the whole processing on the next picture, since the UAV is not moving. According to this strong constraint, the same corners are used to apply the homography to extract the next digit. Thus, processing will be shorter and with a reduced computational cost.

²Red Green Blue

³RANdom SAMple Consensus

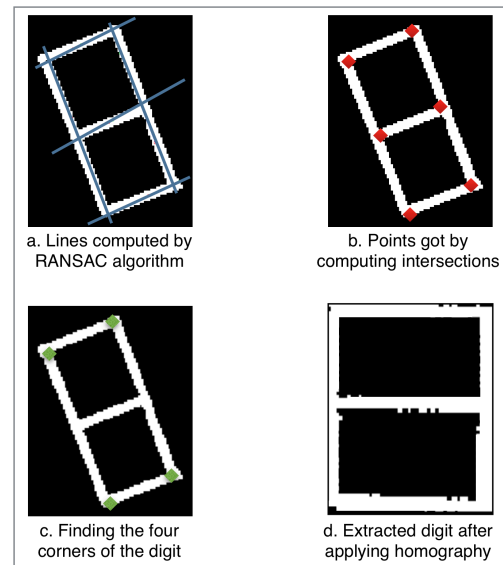


Figure 3: Extraction of the digits bounding box and application of an homography to get an image that does not depend on the viewing conditions

2.3 UAV in stable hovering flight

In this section, the case of the UAV flying in front of the digit panel is considered. Even if the UAV is very stable, it cannot be considered static. Processing presented in section 2.2 cannot work in this case because the embedded camera is a rolling shutter camera. This kind of camera acquires the image one line after the other and as the camera is moving during the process, the different lines are acquired with different camera positions and orientations. Under these conditions, lines for the scenes are not projected in straight lines in the images, and lines generally appear curved due to the rolling shutter distortions. An algorithm of line feature extraction in the image like the RANSAC based one used in the previous section will probably fail. For these reasons, another approach should be used.

The pre-processing detailed section 2.1 is kept as a first step. After this, an edge detector is applied to extract the skeleton of the extracted shape as shown figure 4. This skeleton allows to take out special features as extremities from the shape, intersections of segments and also if there are loops into the shape. With these three features, shown figure 5, it can be determined which digit is represented by the current extracted shape. If these features are not sufficient, extra specific processings can be used to find the correct digit from the shape as shown with the decision tree on figure 6.

There is no guarantee that the decision tree will give the correct answer. But since the computation time is fast enough (a few seconds on a standard laptop), several frames can be processed for the same digit (it changes every thirty seconds).

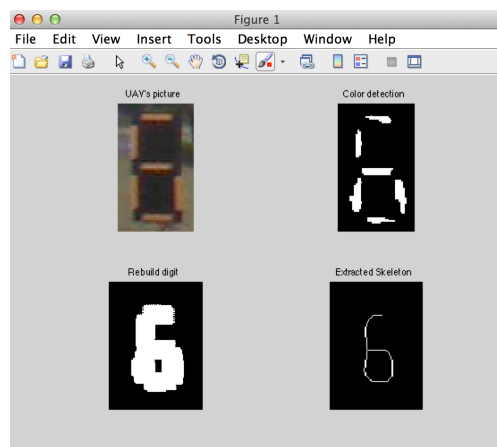


Figure 4: Extraction of the shape's skeleton

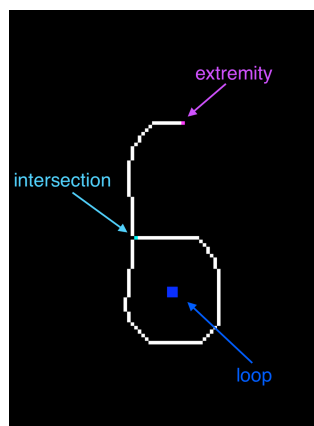


Figure 5: Specific features find out thanks to the shape's skeleton

The final decision can be made after a vote from the all the results.

3 AERIAL MAPPING

The aerial mapping mission consists in producing a mosaic of aerial photographs and then building the map of the village using photogrammetry[2].

3.1 Hardware

To take the aerial photographs, a fixed-wing UAV carries an embedded camera HackHD[7] synchronized with a GPS receiver module and the inertial measurement unit of the UAV. The HackHD camera is a miniature device (see figure 7) with a wide angle lens and that can take pictures with a resolution up to 9 Mega Pixels.

The GPS and IMU are used to determine the position and orientation of the camera while it is grabbing pictures of the

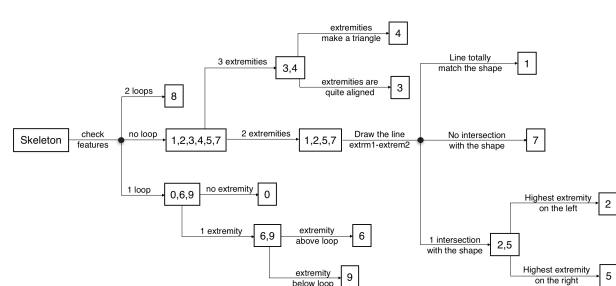


Figure 6: Path followed by the process according to special features extracted to recognize digits

scene. The tight synchronization allows to reduce the computation time of the mapping process by providing near real initial position and orientation of the camera to the photogrammetry algorithm. Also, it allows to create a georeferenced map, id. a map where the correspondence between a position in the map and in the real world is known.

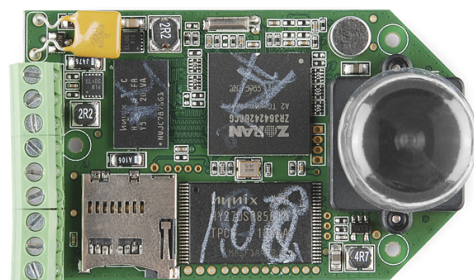


Figure 7: a camera HackHD

3.2 Orthophotomap creation

To create the map, the MicMac[3] program is used as a first step. It is an open-source software developed by the IGN⁴ providing tools to compute tie points⁵, digital elevation model and dense 3D points cloud from a set of overlapping images.

The workflow to compute an orthophotomap with this software is split in several steps:

Tie points computation : first, feature points in the aerial photographs are extracted and then matched between pairs of images (tie points[8] computation),

Bundle adjustment[9] : the 3D position of each feature point and the position and orientation of the camera for each picture is computed,

⁴French National Institute of Geographic and Forest Information

⁵Tie points are feature points from several images that correspond to the same point of the scene

Dense 3D points cloud : afterward, a dense 3D points cloud of the scene using stereoscopy principles is done,

Orthophotomap : finally, the 3D points cloud is projected onto the horizontal plane, and thus the desired orthophotomap is obtained.

However, this approach requires long computation time (about dozens of hours), mainly due to the computation of the dense 3D points cloud. We propose a strategy that allows to shorten the required computation time. From the original photogrammetric process, we replace the third stage by a less precise one which assumes that the environment is locally planar. This allows to avoid the explicit 3D computation of the altitude for the ground at every pixels of the orthophotomap by linearly interpolating it from a simple 3D model named a Digital Terrain Model. The DTM is a model made of a 2D cartesian grid with an associated altitude for each cell.

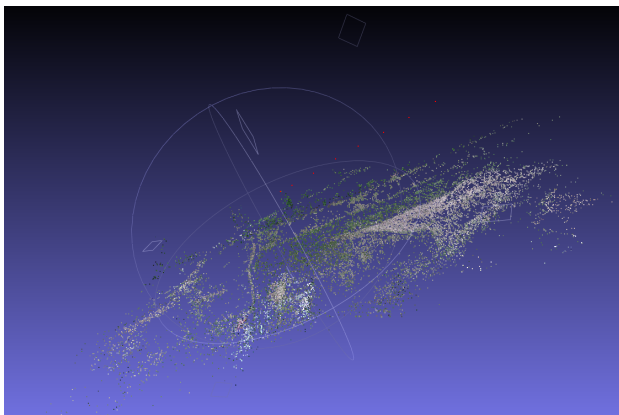


Figure 8: The sparse 3D point cloud obtained from the MIC-MAC tools.

The two first stages of the MicMac software are used to get the sparse 3D point cloud of the feature points. Such a 3D point cloud is shown in the figure 8. A DTM grid is then adjusted to this point cloud by robust least squares fitting. The resulting model, as shown in the figure 9, exhibits a regular mesh structure made of triangles. The robust estimation allows to compute the DTM without corrupting it because of 3D points that were too noisy or even completely wrongly estimated. It also allows to reject the 3D points that were lying on objects above the ground. Finally, the use of the DTM model allows to fill the gaps of the model; if some elements in the grid do not correspond to any 3D feature points, then we interpolate the altitude of the grid element from its neighbors.

Once the grid has been estimated, the generation of the orthophotomap is quite simple. We achieve the same processing for each image grabbed by the camera and then integrate the results. This allows to parallelize by processing different images on different processors and hence reduce the processing

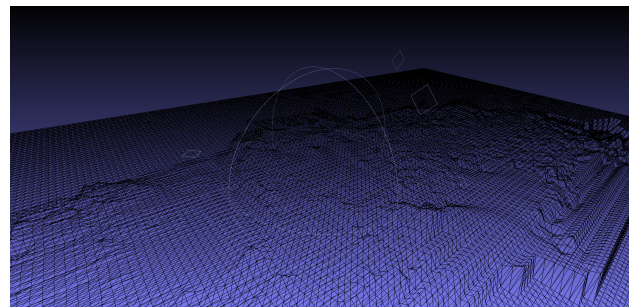


Figure 9: The fitted 3D grid used to model the terrain.

time.

To process one image, each pixel of the associated orthophotomap is projected to the camera, given the position and orientation of the camera and the altitude corresponding to the pixel of the orthophotomap. If this projection lies inside the images boundary, then the orthophotomap is colored from the camera's image pixel. The alpha channel in the image is used to indicate if the pixel of the orthophotomap has been colored or not. The figure 10 shows on image acquired by the camera and the figure 11 shows the corresponding orthophotomap.

Once a sufficient number of images have been processed, a global orthophotomap is generated as shown in the figure 12. As all the orthophotomaps are georeferenced, it is possible to add new data to the global orthophotomap afterwards, for instance to keep the map up to date.



Figure 10: An image from the camera used to construct the map.



Figure 11: The image from figure 10 projected on the 3D map.



Figure 12: The result of the projection of ten images on the 3D map.

4 CONCLUSION

In this paper, the methods for solving two of the IMAV2014 mission elements are presented. The first one concerns an automatic digit detection and recognition. Since all the aspects of the mission are not known, several algorithms are investigated in order to get the most robust solution on the day of the competition. The second mission element consists in making a map of the flight area, by stitching several pictures. An off-line solution has been developed using images and the associated 3D poses acquired by GPS and IMU. The proposed solution is using bundle adjustment in order to improve the alignment of the images. As processing time is a key concern during the competition, a compromise has been made. The proposed solution provides better results than simple 2D images mosaicking and it doesn't require as much time as the complete 3D orthophotomap generation thanks to the interpolation of altitudes in the DTM model.

REFERENCES

- [1] Andrea Fusiello Roberto Toldo. Robust multiple structures estimation with j-linkage. <http://www.diegm.uniud.it/fusiello/papers/eccv08.pdf>, October 2008.
- [2] T. Schenk. Introduction to Photogrammetry. <http://www.mat.uc.pt/~gil/downloads/IntroPhoto.pdf>, 2005.
- [3] IGN. MicMac. <http://logiciels.ign.fr/?-Micmac>.
- [4] Mutsumi Omori Tetsushi Koide Masaharu Yamamoto, Anh-Tuan Hoang. Speed traffic-sign recognition algorithm for real-time driving assistant system speed traffic-sign recognition algorithm for real-time driving assistant system speed traffic-sign recognition algorithm for real-time driving assistant system. http://sasimi.jp/new/sasimi2013/files/pdf/p195_R3-11.pdf.
- [5] A.Srisaila S.Kranthi, K.Pranathi. Automatic number plate recognition. <http://digitalpdf.org/automatic-number-plate-recognition-dov> July 2011.
- [6] Kenney-C.S. Manjunath B.S. Zuliani, M. The multi-ransac algorithm and its application to detect planar

homographies. <http://vision.ece.ucsb.edu/publications/05ICIPMarco.pdf>, September 2005.

- [7] HackHD camera. <http://www.hackhd.com/>.
- [8] Kourosh Khoshelham. Role of Tie Points in Integrated Sensor Orientation for Photogrammetric Map Compilation. http://www.asprs.org/a/publications/pers/2009journal/march/2009_mar_305-311.pdf, march 2009.
- [9] Richard Hartley Bill Triggs, Philip McLauchlan and Andrew Fitzgibbon. Bundle Adjustment — A Modern Synthesis. <http://lear.inrialpes.fr/pubs/2000/TMHF00/Triggs-va99.pdf>, 2000.

Flexible framework for the development of versatile MAV systems for multi-disciplinary applications

M. Cordero*, M. A. Trujillo, J. Ruíz, A. Jiménez, L. Díaz, and A. Viguria

Center for Advanced Aerospace Technologies, Wilbur y Orville Wright 19, La Rinconada (Seville, Spain)

ABSTRACT

MAV (Micro Aerial Vehicle) systems have proven useful in multiple applications (e.g. aerial photogrammetry, aerial inspections,...). For each application different sensors and even different airframes are often needed. Having a flexible UAS (Unmanned Aircraft System) devices (specifically an autopilot, a payload manager and a ground control station) that can be seamlessly used with different platforms is of great interest as it reduces the development time and effort (and hence cost). This flexibility must apply to the hardware, software and design methodologies so the system can be rapidly adapted to meet the requirements of other applications. This paper introduces a flexible system developed by CATEC consisting of an autopilot, Ground Control Station (GCS) and a payload manager for MAVs. This system has already been tested in real experiments for different applications that are also presented in this paper.

1 INTRODUCTION.

Unmanned aircrafts are finding an increasing number of applications such as precision agriculture, firefighting, mining, search and rescue, surveillance,... to cite a few. Depending on the mission requirements, different aircraft systems may be needed (e.g. more autonomy may be needed for the inspection and mapping of wider areas). Furthermore, different sensors need to be used in different applications (e.g. multispectral cameras are very useful in precision agriculture applications but may not be needed in search and rescue missions). Traditionally, UAS (Unmanned Aircraft systems) have used their own ad-hoc systems (i.e. autopilot, payload manager, Ground Control Station (GCS), etc). These systems are specific for their aerial platform and cannot be easily adapted to be used in another one. Dealing with multiple different UAS systems increase operating and maintenance costs. It is of great interest to have flexible and versatile autopilot, payload manager and ground control station that can be used with different aerial platforms and for different missions. In the last decade different autopilots and GCS systems that can be easily integrated in different aerial platforms

have been developed. Most of these autopilots are COTS (Commercial Off-The-Shelf) products with a very limited access to their internal operation (e.g. Piccolo from Cloud Cap Technologies [1], MP2x28 from MicroPilot [2],...). There are also open source autopilots that provide full access to the autopilot's internals. The most popular ones are probably Ardupilot [3] and Paparazzi [4]. These autopilots implement the GNC (Guidance, Navigation and Control) algorithms directly in C++ code. This can be tricky for some control engineers not used to work with programming languages. In this paper, a flexible and versatile UAS system developed by CATEC is presented. This system includes an autopilot, a GCS and a payload manager. Regarding the autopilot, the control engineers follows a model-based development (MBD) approach in which the GNC algorithms are developed in Simulink and ANSI C code is obtained using an automated code generation tool from TheMathWorks. The GCS and payload manager also use flexible architectures as will be shown. This paper is organized as follows: in section 2 a description of the system and the individual components (i.e. autopilot, GCS and payload manager) is provided; in section 3 the flexibility and versatility of this system is illustrated by showing three applications in which the system has already been used; finally in section 4 the conclusions are presented.

2 SYSTEM ARCHITECTURE.

The proposed system architecture is depicted in Figure 1. The elements of the ground and the airborne segments are shown in red and blue respectively. Additionally, these elements can be divided in those that are used for GNC tasks and those that are used for mission specific tasks. The autopilot, GCS and navigation sensors belong to the first group while the payload sensors, payload manager and payload ground software belong to the second group. The elements of the ground and the airborne segments communicate using two radiolinks operating at different frequencies (e.g. 900 MHz for the command and telemetry link and 2.4 or 5.8 GHz for the payload datalink). It is important to note that the elements of the navigation and mission payload groups are not isolated. Instead, some of them are interconnected (e.g., the autopilot is connected to the payload manager to provide it with telemetry such as the position, velocity and attitude of the aircraft that can be exploited for georeferencing sensor data).

*Email address: mcordero@catec.aero

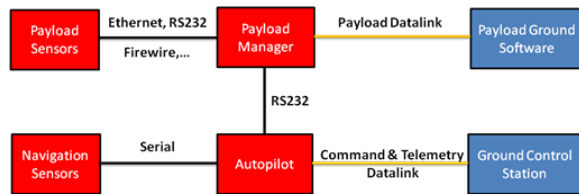


Figure 1: System architecture developed for the precision agriculture application.

2.1 Autopilot.

The autopilot controls the throttle and control surfaces to safely fly the aircraft according to the mission configured by the operator by means of the Ground Control Station (e.g. following a given path of waypoints). For this purpose, a navigation module is used to process the information provided by the navigation sensors (i.e. inertial sensors, magnetometers, GNSS (Global Navigation Satellite System) receiver and differential pressure sensors) to estimate the aircraft state (i.e. position, velocity and attitude). From this estimation of the aircraft state and the established mission, the control signals are calculated and sent to the aircraft servos and engine throttle. CATEC's autopilot is divided in two boards: a high-level board (or control board) and a low level board (or safety board). The control board is based in an ARM Cortex A8 core running at 720 MHz with 256 MB of RAM and runs the GNC (Guidance, Navigation and Control) algorithms over QNX operating system. The safety board is in charge of collecting the data from the navigation sensors and sending the control signals to the actuators. This board is based on an ARM Cortex M4 microcontroller running at 80 MHz using FreeRTOS as the operating system. This board also implements safety features including a light navigation and control module that allows the aircraft to safely fly according to a return to home mission. The autopilot also includes the interfaces for communicating with other systems (e.g. a radiolink for telemetry, a RS232 interface for connecting with the payload manager,...) and PWM (Pulse Width Modulation) outputs that to control the aircraft control surfaces. The functional and hardware architecture of the autopilot are shown in Figure 2 and Figure ?? respectively.

The autopilot software components that implement the GNC algorithms have been developed following a model-based development approach using Simulink [5]. This approach is based on automatically generating standard ANSI C code from the Simulink models. This code can be compiled for QNX or FreeRTOS and run in an embedded PC. With this framework the development workflow can be accelerated as you can develop a model with the GNC algorithm, generate code and run it in the embedded PC and check its behaviour and performance. When an error is detected or any change needs to be applied the Simulink model can be modified and the code can be generated again to repeat the iterative process

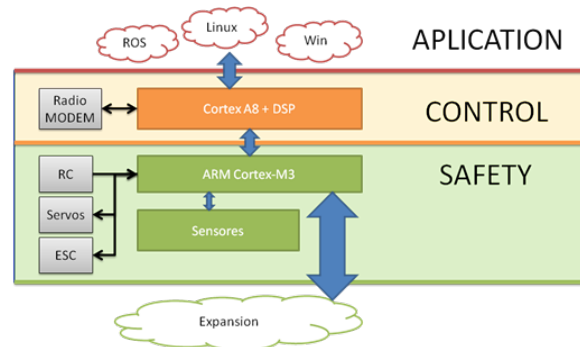


Figure 2: Autopilot functional architecture.

of develop-and-test until everything works as expected. For monitoring the behaviour of the system, Simulink External Mode is a extremely helpful tool. This tool provides communication between the Simulink model and the running generated code so any signal from the model can be monitored e.g. using a scope or logging the data for postprocessing.

2.2 Ground Control Station (GCS).

The GCS software runs in a laptop PC and is used by the aircraft operator for commanding waypoint paths, monitoring telemetry, logging flight data and managing alarms. CATEC's GCS has been developed using Qt open source frame work in C++ programming language. A Model-View-Controller (MVC) software architecture is used to decouple the code components related to the data model from the code components of the graphic part and from the internal software logistic. The functionality of the set of basic libraries that conforms the core of the GCS can be easily expanded thanks to a plug-in system. A number of plug-ins is also provided together with a simple API so third-party developers can use their own plug-ins. A plug-in manager is in charge of loading/unloading all of them. The following plug-ins are available:

- States plotter. This plug-in plots the received telemetry data for debugging and monitoring purposes.
- Logger. This plug-in logs the received telemetry data in a file in CSV format so it can be processed later.
- Primary Flight Display. This is a graphic plug-in developed in GLStudio (with a Qt/C++ wrapper) that shows the aircraft attitude, speed and flight altitude.
- Payload Remote Manager. This plug-in allows the configuration of the payload sensors from the ground. This will be analyzed in more detail in the following section.
- Waypoint Editor. This plug-in is used for the edition of paths of waypoints that are uploaded to the aircraft autopilot.

Finally, the View component is the one that shows the graphical interface for the aircraft operator. It is completely independent from the core GCS software so different view components can be easily interchanged. The layout can be easily modified, docking and undocking the different components of the view. Figure 3 shows the graphical user interface of CATEC's GCS.

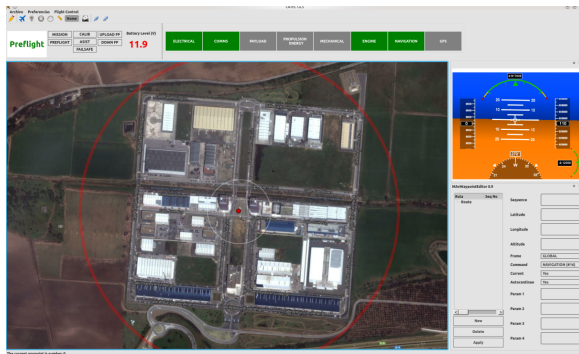


Figure 3: GCS graphical user interface.

2.3 Payload manager.

The payload manager is in charge of managing the different elements that are part of the aircraft payload. It performs multiple tasks including sensor data collection, sensor configuration, status monitoring and it also manages the communications with the GCS. As mentioned in the previous section, the GCS is used for controlling the sensors operation during the flight by sending commands to them via a radiolink. These commands are received by the payload manager which processes them and performs the associated actions. These commands can for example start or stop the capture of video data, change the frame rate and/or image resolution, activate or deactivate the transmission of the collected data to the GCS, etc. The flexibility of the system makes it easy to implement new configuration commands. The payload manager also monitors the status of the different elements of the payload and reports them to the GCS. The collected data can be locally logged in flash memory and/or can be sent to the GCS during the flight by means of a radiolink. Additionally, some data processing can also be performed onboard before logging or sending them to the GCS. For example, is a common practice to add a tag to the collected data with the state of the aircraft (i.e. position, velocity, attitude, etc) in the instant when that data was collected and a time stamp. With this information, collected data can be exploited e.g. for mosaicing, DEM (Digital Elevation Model) generation, etc. Depending on the mission and the application, different sensors can be needed on board the aircraft e.g. visual and infrared cameras, multispectral cameras, LIDAR (Light Detection and Ranging), sonar sensor, etc. The system has to be flexible both in hardware and software for being able to manage the different

sensors and performing the different tasks of each mission.

Regarding the hardware the payload manager developed by CATEC is based on a Falcon VL-EPU-2610 embedded PC board from VersaLogic [6]. This board includes an Intel Atom E6x0T processor running at 1.6 GHz, 2 GB RAM module and multiple interfaces including four USB ports, one serial RS-232 port and one Gigabit Ethernet interface. It also includes a miniPCI slot so new interfaces can be added easily. Figure 4 shows the payload manager installed onboard an autonomous helicopter next to an infrared camera.

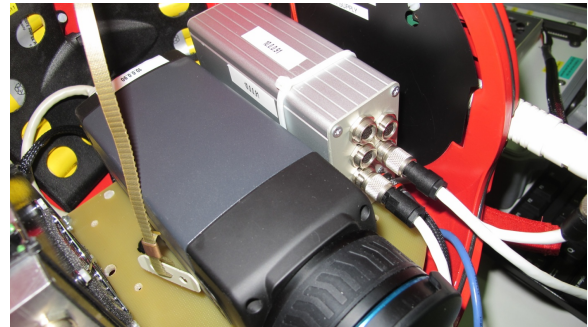


Figure 4: Payload manager system and IR camera on-board an unmanned helicopter.

Regarding the software architecture, the payload manager is entirely based in ROS (Robotic Operating System) and Qt and runs over a Linux distribution. ROS is an open source framework widely used in the robotics community and increasingly in the industry. This framework facilitates the development of modular applications so different modules (which are referred to as nodes in ROS terminology) can be developed independently and added for new functionalities. The Qt framework has been selected for its large set of multi-platform tools for handling files, threads, processes management, UDP/TCP communications, etc. Communication between ROS nodes is based in the publish-subscribe paradigm. This paradigm is based on the definition on different data topics (e.g. images, telemetry data, node status, etc). The different existing ROS nodes can publish on these topics and/or subscribe to them for sharing information. Figure 5 shows the payload manager software architecture. Red nodes are part of the payload manager and hence run onboard the aircraft while the blue node is the payload manager plugin of the GCS. The system includes the following nodes:

- IR camera node is a mission specific node for infrared cameras. It handles the connection with the IR camera via an Ethernet connection satisfying the GenICam standard. This node publishes an IR-image topic each time the camera captures a thermo-graphic image. It also provides some services for changing some of the IR camera parameters so other nodes can change them.
- Visual camera node manages webcam devices compat-

ibles with the video4linux driver. This node publishes an RGB-image topic each time the camera captures an image. It also provides services for changing some parameters of the camera such as brightness, contrast, FPS (Frames per Second), etc.

- Streamer node subscribes to the image topics of any sensor node (e.g. the IR and/or visual cameras) in order to transmit them to the GCS software via an UDP connection and a wireless link. This node can use different video codecs for compressing the images in a video stream in order to reduce the bandwidth usage. The stream sent to the GCS can consist in one or more video streams and it can also include a metadata stream with additional information of each frame.
- Telemetry node accesses the telemetry data provided by the autopilot via a serial connection. For each data acquisition a telemetry topic is published with information related to the platform position and attitude. This node doesn't provide any service.
- Logger node doesn't publish any data topic but it subscribes to all the data topics provided by other sensor nodes in order to log them in a database file (shown in green in Figure 5). It provides basic services for starting or stopping the logging of topics. After each logging procedure, a BAG file is stored in the storage unit of the embedded system. BAG is a ROS-specific format for compressed data that can be easily read and processed with the tools provided by ROS.
- Comms node manages the communications with the GCS payload manager software. On one hand, it implements a TCP server that handles incoming connections from the GCS for listening to operational commands. These commands are then translated to the specific service requests to the target ROS nodes. In the other hand, the node is sending periodically the status of the nodes to the GCS payload management software.
- Monitor node collect the status topic published by all the other nodes. This topic indicates if they are working properly or not. The purpose of the monitor node is to launch the others, monitor their status and if something goes wrong restart them.

ROS nodes implementing the interfaces of topic publications, subscriptions and services defined in Figure 5 could be immediately replaced by other nodes with the implementations for different hardware devices providing the system with high flexibility.

3 APPLICATIONS.

CATEC's system has been tested in several applications proving its versatility. In this paper three of these applications

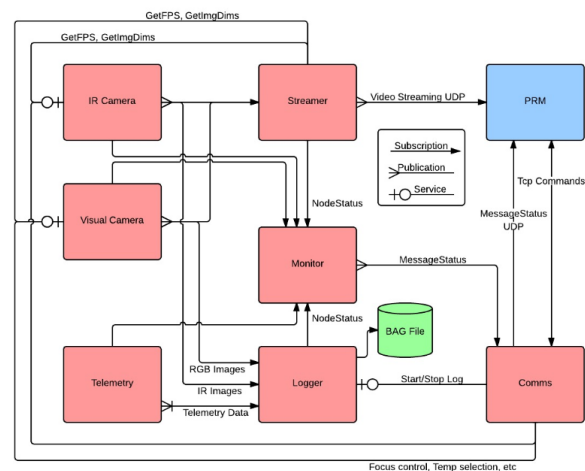


Figure 5: Application software modules. Red nodes run onboard the UAV and blue ones run in the GCS payload management software.

(aerial robotics manipulation, precision agriculture and wind blade inspections) are presented to illustrate the flexibility of the system.

3.1 Aerial robotic manipulation.

This work was developed as part of the European Commission's 7th Framework Programme (FP7) project ARCAS [7]. The objective of this project is to integrate robotic manipulators onboard of aerial robots for structure assembly tasks. In the indoor tests, a robotic arm developed by CATEC was integrated with an octocopter. CATEC's autopilot and payload manager were both integrated onboard the octocopter. During the indoor flights, GNSS signals are not available so positioning information was obtained through a motion capture system by Vicon which is depicted in Figure 6. This system is based on the real-time processing of the information provided by 20 Vicon cameras that can provide the position of an object with sub-millimetric accuracy. The system only needs that passive markers are attached to the objects to be tracked. The positions are calculated by a Vicon central computer and sent to the autopilot in real time by means of a WiFi connection. The ability to substitute the GNSS positioning by Vicon positioning for indoor flights shows the flexibility of the CATEC's autopilot. The payload manager performed the already mentioned tasks of sensor data acquisition, logging and streaming and sensor configuration. In addition, the payload manager included a ROS node that implemented a high-level controller that provided the autopilot and robotic arm controller with position references for the octocopter and the robotic arm respectively. This way, difficult tasks such as coordinated control of both the aerial vehicle and the onboard robotic arm can be achieved.

Two different experiments were carried out. In the first



Figure 6: Vicon camera (left) and CATEC indoor testbed (right).

one, the octocopter had to pick up a bar estimating its position from four visual markers that were attached to it as shown in Figure 7. A visual camera was integrated onboard together with a light source to guarantee good illumination conditions for the vision algorithm to work properly. In the payload manager, a ROS node was in charge of the acquisition of the visual data and another ROS node processed this data conveniently to provide the autopilot and the robotic arm controller with position references. With this architecture the final task of autonomously picking up the bar with information from the visual camera was achieved successfully.



Figure 7: Grabbing a bar with visual markers.

In the second experiment the octocopter had to autonomously grab three bars with special connectors and sequentially assemble them to form a structure. In this case, Vicon markers were attached to the bars so the Vicon system calculated their attitudes and positions and sent them to the ROS node implementing the high-level controller. As in the

previous case this node provided the autopilot and robotic arm controller with position references. This experiment was also conducted successfully. Figure 8 shows two frames of the video that was recorded during the flight trials.

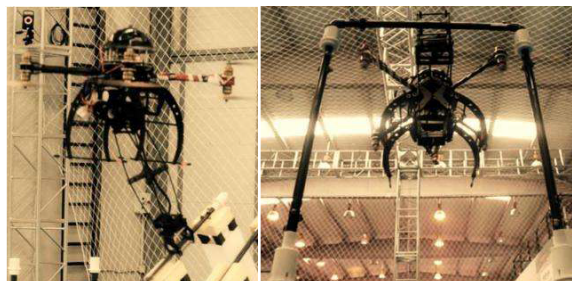


Figure 8: Octocopter grabbing a bar (left) and assembling the structure (right).

3.2 Precision agriculture.

This work was developed as part of the European Commission's 7th Framework Programme (FP7) project Field-copter [8]. The objective of this project was to investigate the added value of Unmanned Aerial Systems (UAS) over satellite-based remote sensing. The principal benefits of UAS are their flexible deployment at any time in the day and their ability to acquire imagery under cloudy conditions. An infrared and visual camera were integrated onboard an unmanned helicopter (specifically a SARAH UAV manufactured by Flying-Cam). It is a completely electrical helicopter with 7 kg payload capacity. It has around 20 minutes of autonomy and its operation is completely autonomous, including taking-off and landing. This UAV uses a commercial autopilot so only CATEC's payload manager was integrated together with visual and infrared cameras as was shown in Figure 4. Additionally, CATEC's GCS was used for payload management purposes. The experiments were conducted in vineyards in Lleida (north of Spain). In this case, the payload manager stored the collected visual and IR images in the onboard storage device. After the flights, the data were post-processed to obtain mosaic images and segmentation of the individual vines as shown in Figure 9.

3.3 Wind blade thermographic inspection.

Wind turbines are gaining importance in the last years because of their high efficiency during energy production without greenhouse gas emission. However, wind blades maintenance operations have become expensive due to operating costs (accessibility, scaffolding,...) and the cost of having them unemployed. A non-destructive technique capable of detecting most significant in-service defects in composite wind blade is Infrared Thermography (IRT). In this work, a feasibility study for defect detection during maintenance operations in wind blades using unmanned aircrafts was performed. This approach allows inspecting the blades without

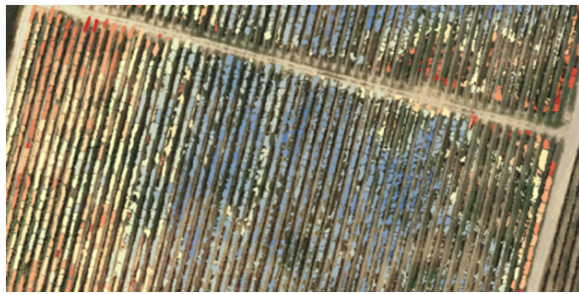


Figure 9: Canopy temperature obtained from flight experiments showing the obtained object-segmentation of the individual vines.

any scaffolding system and reducing the time they remain unemployed.



Figure 10: UAV flights close to the wind turbine to be inspected (left); Trajectory followed for the inspection (right).

In this case, a similar configuration to that used in the precision agriculture experiments was adopted. An infrared camera was integrated onboard the SARAH unmanned rotorcraft. To obtain an adequate image resolution and to keep a safe operation area for the UAS, a distance of about 5 meters from the wind turbine blade and displacement speed of 1 m/s were defined. The inspection strategy (shown in Figure 10) was designed to achieve the entire blade inspection using a single battery (whose autonomy is around 20 minutes).

The acquisition and logging of the infrared images was controlled by CATEC's payload manager and monitored with the payload manager plug-in of the ground control station. The feasibility of the proposed inspection methodology was demonstrated for detecting the most common wind blade service defects such as cracks, delaminations and impacts, by using passive infrared thermography. Figure 11 shows some of the thermographic images obtained during the flight trials performed in Tarifa (south of Spain) in which 10-meter blades were inspected.

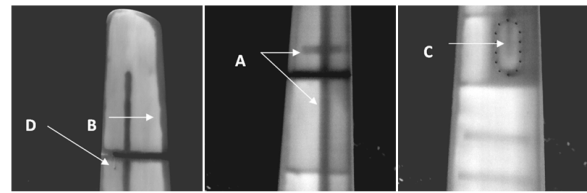


Figure 11: Thermal images showing (A) pitch brake system, (B) bonds between upper and lower shell, (C) access gate and (D) surface roughness.

4 CONCLUSION.

In this paper, the UAS systems developed by CATEC for unmanned aircrafts have been presented. These system have been developed following a flexible architecture both in hardware and software. This allows them to be easily adapted for their use in different missions and applications with different requirements, as well as being integrated in different aerial platforms. The systems include an autopilot, a payload manager and a ground control station with remote payload management functions. These have been used in real flight demonstrations for different applications. Three different applications have been shown, namely aerial robotic manipulation for structure assembly, precision agriculture and wind blade termographic inspection.

REFERENCES

- [1] CloudCapTechnologies. Piccolo II Datasheet.
- [2] MicroPilot. MP2x28 Series Autopilot <http://www.micropilot.com/products-mp2028-autopilots.ht>.
- [3] Ardupilot. <http://ardupilot.com/>.
- [4] Paparazzi UAV Autopilot. http://wiki.paparazziuav.org/wiki/Main_Page.
- [5] Daniel Santamaría, Francisco Alarcón, Antonio Jiménez, Antidio Viguria, Manuel Béjar, and Aníbal Ollero. Model-Based Design, Development and Validation for UAS Critical Software. *Journal of Intelligent & Robotic Systems*, 65(1-4):103–114, August 2011.
- [6] VersaLogic. Falcon (VL-EPU-2610) <http://www.versalogic.com/products/ds.asp?productid=2>.
- [7] ARCAS. <http://www.arcas-project.eu/>.
- [8] Fieldcopter. <http://fieldcopter.eu/>.

Design of a Quadrotor System for an Autonomous Indoor Exploration

M. Gäbel,* T. Krüger and U. Bestmann
University of Braunschweig – Institute of Technology

ABSTRACT

This paper describes the hardware components and the embedded software of a self-designed unmanned aerial system (UAS) to fulfill the task of an autonomous indoor exploration. The project, the paper is based on, was conducted at the University of Braunschweig - Institute of Technology by students and scientific researchers studying and working at the Institute of Flight Guidance. The paper is structured as follows. The first part deals with the development, selection, collocation and assembling of the appropriate hardware to obtain a reliable and robust system which suits the task of controlling the UAS, collecting information about the environment and providing the computational resources to execute the implemented algorithms. The second part deals with the software architecture that generates the control commands for the quadrotor UAS and ensures the execution of the mission by the stepwise fulfillment of single tasks.

1 INTRODUCTION

Investigating the interior of a building with UASs has gained interest during the past years. Especially the use of multirotor systems such as quadrotor UAS has been taken under consideration by many researchers and is still topic of current studies as they are highly maneuverable and thus especially suited to fulfill challenging indoor tasks. Moreover, compared to helicopters and flapping wings, the design of multirotor systems allows to carry a wide range of scientific payload. Several studies, experiments and competitions [1],[2],[3],[4],[5], [6] have been undertaken and carried out to analyze and present the suitability of multirotor systems for indoor and outdoor tasks. As a result of the promising findings there is ongoing research to develop and implement new hardware and software to extend the degree of autonomy, the sensing abilities and the flight duration of multirotor systems. Especially in the case of a natural disaster UASs could be used to obtain an overview of the situation inside and around buildings to assist rescue units. This paper introduces an approach to deploy a quadrotor UAV for the task

of an autonomous indoor exploration. The proposed quadrotor system is equipped with the Pixhawk autopilot, a Hokuyo laser scanner, an Optical Flow Sensor and an onboard computer with a low power consumption.

2 HARDWARE

In most cases for the purpose of doing scientific research including the operation of multirotor UAS the Astec Pelican is considered first as it satisfies the most users demands regarding scientific payload, endurance and computational capacity provided i.e. by the Astec Mastermind. Nevertheless for our demands the Astec Pelican has proven to get close to its limits regarding the payload and flight endurance. Therefore we designed and built a quadrotor UAS from scratch, which is able to carry the required payload, provides sufficient computational resources for the onboard calculations and guarantees a minimum flight time of 25 minutes. The quadrotor UAS with all its components and an detailed overview about how the single components are linked with each other is represented in Figure 1. In its final configuration the quadrotor UAS weights about 3 kilograms and has a maximum diameter of 76 centimeters.

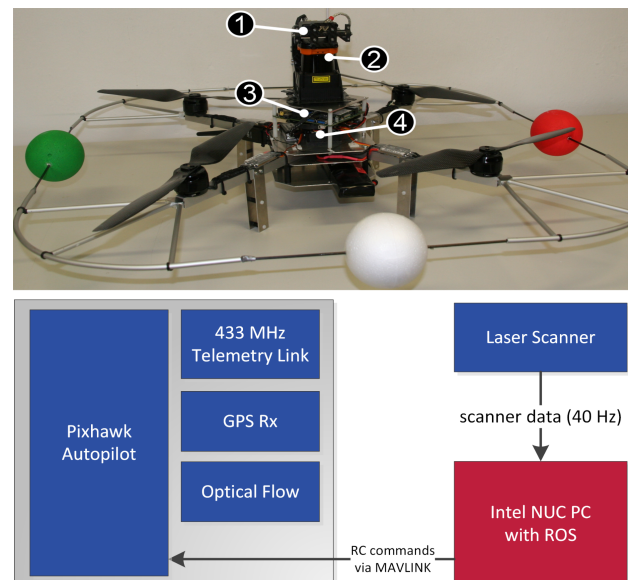


Figure 1: Quadrotor UAS

*Email address(es): m.gabel@tu-braunschweig.de, th.krueger@tu-braunschweig.de, u.bestmann@tu-braunschweig.de

2.1 GNSS Unit

The GPS unit is mounted on top of the laser range finder and shown in Figure 1 marked with number 1. It consists of an ublox LEA-6H satellite receiver and the Honeywell HMC5883L 3-axis digital compass. The ublox satellite receiver is capable of handling signals from the Russian GLONASS or the American GPS and is able to achieve a horizontal dilution of precision of 2.5 meters assuming optimal conditions. Furthermore the ublox LEA-6H satellite receiver is able to process the signal of a satellite based augmentation system to enhance the accuracy of the calculated position solution up to a horizontal dilution of precision of 2 meters. The update rate of the position can be modified and set to a frequency of 1 or 5 Hz. Right after starting the satellite receiver it takes about 26 seconds to obtain a first position solution. In case the GNSS unit has just been switched off for a few seconds (hot start) or a signal of a satellite based augmentation system (SBAS) is available a first position can be obtained within a few seconds. The built in Honeywell magnetometer guarantees a compass heading accuracy with a maximum deviation of 2 degrees. Either devices, the GNSS receiver and the Honeywell magnetometer, are supplied with 3.3 Volts by the Pixhawk autopilot.

2.2 Hokuyo UTM-30LX

The Hokuyo UTM-30LX is shown in Figure 2 and marked with the number 2. The laser range finder device is mounted on the quadrotor UAS to sense the surrounding environment of our quadrotor UAS. With a scanning range from 0.1 up to 30 meters, a field of view of 270 degrees covered within 25 milliseconds, a weight of 210 grams and a power consumption of only 8 Watts it is the main sensing device. Intending to travel at a constant altitude that should not be occupied or blocked by free standing obstacles, i.e. furniture, the implementation of a device sensing obstacles just in a two-dimensional plane is suitable to fulfill the task of exploring the interior of a building while staying at the same floor. Besides the purpose of sensing obstacles, the information gathered by the laser range finder are required to perform the simultaneously localization and mapping (SLAM) and to reveal the unknown space inside the building.



Figure 2: Hokuyo UTM-30LX

2.3 Intel NUC i5 Board

The multirotor system has been equipped with the Intel D54250WYB NUC board shown in Figure 3 a). With its 4th generation Intel Core i5, 120 GB SSD harddrive and 8 GB of memory it provides the required computational power to perform the high-level task such as localization, mapping, obstacle detection, path- and motion planning and moreover provides enough disk space to log data and install the required software. Further advantages of the NUC board are its compact dimensions of 11 x 11 x 3 centimeters (w x b x h) and the low required power consumption within the range of 9 Watts at idle and a maximum of 24 Watts at full load. In addition the NUC board is equipped with a wireless local area network (WLAN) module with integrated Bluetooth to establish a reliable link between the onboard PC and the ground control station outside the building. Mainly, the link serves the purpose to monitor the onboard processes and visualize the progress of the mission.

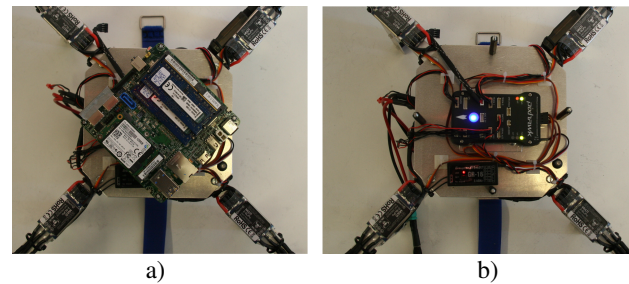


Figure 3: Image conversions

2.4 Pixhawk autopilot

The Pixhawk autopilot, depicted in Figure 3 b), is a state-of-the-art autopilot equipped with the quite powerful STM32F427 Cortex M4 168 MHz processor with 256 KB of RAM and a flash memory of 2 MB. In addition to the main CPU the autopilot is equipped with a STM32F103 processor to avoid the loss of the UAV in case the main processor fails. The firmware of the Pixhawk autopilot, which can be set up with the open source ground control QGroundcontrol, can be adjusted for different kinds of UASs. Furthermore the autopilot's architecture allows to replace the original Pixhawk firmware by another compatible autopilot firmware i.e. Ardupilot. The autopilot itself integrates a 16 Bit gyroscope, a 14 Bit accelerometer and magnetometer both manufactured by STMicroelectronics and a MS5611 barometer which can be used as an altimeter and variometer with a minimum resolution of 10 centimeters. To extend the degree of autonomy the autopilot provides several interfaces to connect additional sensors like an optical flow sensor, airspeed sensor or a GPS unit via USB, SPI (Serial Peripheral Interface), CAN (Controller Area Network) and I²C (Inter Integrated Circuit). In addition there are interfaces for two telemetry modules and a Spektrum DSM receiver.

3 SOFTWARE

The developed and implemented software shall meet the major task to guarantee the safe operation of the quadrotor UAS at every point of the mission. Since this task must be achieved in a limited amount of time, at first a look around for applicable open source projects in the field of robotics was conducted. As a result the Robot Operating System (ROS) has been considered as a candidate to push the project forward. The Robot Operating System has been developed at Stanford Artificial Intelligence Laboratory for the Stanford-AI-Robot (STAIR) project in 2007 and evolved since then to a platform suitable for a wide range of robotic applications. Today ROS's further development is driven by Willow Garage¹ and the nonprofit organization Open Source Robotics Foundation (OSRF)². The success of ROS is based on the large number of industrial, scientific and common co-contributors and the quite active community. Furthermore the modular design of the available software packages and the corresponding tutorials facilitates the first steps into ROS and the implementation of the algorithms required for the own project. At the moment there are more than 3000 packages available ranging from SLAM- and navigation algorithms to camera and laser drivers. In addition ROS integrates also external libraries like OpenCV or the PointCloud Library (PCL). Since ROS already provides drivers for several devices and moreover provides additional software with a variety of functionality, we decided to use ROS and to integrate the developed software in the typical ROS manner as standalone packages. The Robot Operating System and the software packages run on the NUC board described in the hardware section. The implemented software consists of the algorithms performing laser data processing, SLAM, exploration, path planning, path tracking, mission control and the communication interface between ROS and the Pixhawk autopilot. A flow chart for the exploration task integrating the hardware and the algorithms is illustrated in Figure 4. The essential algorithms and their interactions will be explained in the following subsections.

3.1 SLAM

For the task of a precise indoor navigation a robust localization of the robot is quite essential. Overall the UAS is equipped with three different kind of sensors that could be used for this task. The mounted optical flow unit is directly connected to the autopilot to provide complimentary information about the current velocity and altitude of the quadrotor UAS in its body axis system. Thus the sensor data of the optical flow unit might be already out of time due to the transfer latency when processed by an algorithm running on

¹Willow Garage was founded 2006 by Scott Hassan "to accelerate the development of non-military robotics and advance open source robotics software." <http://www.willowgarage.com/>

²"The mission of OSRF is to support the development, distribution, and adoption of open source software for use in robotics research, education and product development." <http://osrfoundation.org/>

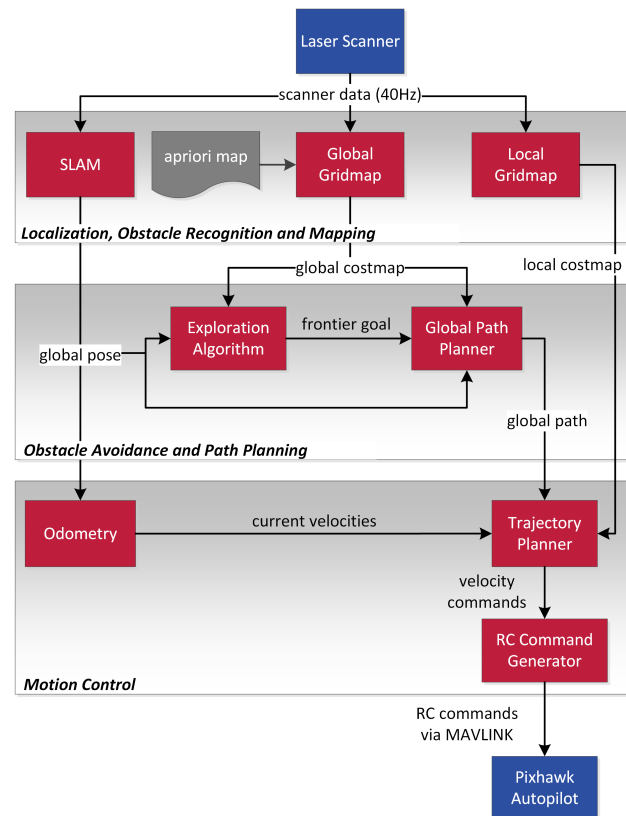


Figure 4: Exploration software configuration

the NUC board. The IMU is not appropriate for the task as well since suffering the same problem besides its biased measurement results. Therefore another approach has been taken into consideration. As the quadrotor UAS is equipped with the Hokuyo laser range finder directly connected to the NUC board, a SLAM [7] algorithm has been implemented to localize the robot and recognize its movement while mapping the environment. To fulfill this task the Hector SLAM algorithm [8] has been implemented and extended to provide additional information about the linear velocities and angular rate in the x-y plane of the 3 dimensional, global coordinate frame. Among the available LiDAR-based (Light Detection and Ranging) SLAM algorithms the Hector SLAM algorithm is preferred since the algorithm is open source and already implemented in ROS as a standalone package. The Hector SLAM estimates the position and orientation of a UAS by trying to match the incoming laser scans. Therefore the Scanmatcher algorithm aligns the latest laser scan with the map generated by all the previous laser scans. An example of a map generated by the Scanmatcher is shown in Figure 5. In order to work, compared to other SLAM algorithms like GMAPPING, the Hector SLAM algorithm does not require any odometry information. In addition the algorithm

can be even used on 6 DoF platforms whereas others like the GMAPPING algorithm, which do not take into consideration pitch and roll angles, might fail without modifications of the underlying code. Moreover the Hector Slam algorithm is able to cope with the high publish rates of the Hokuyo laser range finder. The Hector SLAM algorithm is a typical front-end SLAM, which focuses on estimating the robot movement in real-time and does not make use of any kind of pose graph optimization, i.e. loop closing. Nevertheless the algorithm has proven to work reliable and accurate in many small scale scenarios and therefore it has been chosen to perform the 3D localization of the UAS in the unknown environment inside the building. For the task of indoor exploration the 3D pose estimate is passed to the exploration algorithm and both gridmaps. Furthermore the odometry algorithm keeps track of the global pose changes to compute the current velocities of the UAS. The map generated by the Hector SLAM algorithm is only used to verify the overall performance of the SLAM algorithm. The mapping result itself is not fed into the costmaps, which are only updated by the incoming laser scanner data. The algorithms generating the maps, the scanmatcher map and the gridmaps, can be adjusted to filter the laser scanner data in terms of range measurements. Thus the information stored in the scanmatcher map and global gridmap can differ from each other depending on the range limits for the corresponding filter.



Figure 5: Scanmatcher map

3.2 Exploration

The implemented 2D Exploration algorithm has been proposed for the first time by Brian Yamauchi in 1997 [9]. To suit the defined objectives the grid-based exploration algorithm is initialized with an apriori map of the building. To obtain a

suitable map for the indoor exploration task an image of the building's contour is required. The image is manipulated in such a way that only the interior of the building will be considered unknown which will force the robot to explore only the inside of the building. The SDL (Simple DirectMedia Layer) library is used to convert the manipulated image into a binary file containing the gray scale information. Incorporating the binary file, the information about the image resolution, the size of a grid cell and the gray scale thresholds, each cell of the grid is categorized into free, occupied or unknown. The result of combining all these information is a scaled grid-map also called occupancy grid. An example of the steps for generating such a map is depicted in Figure 6. Figure 6 a) presents the top view of our Institute which is used to derive a simple image of the Institute's shape shown in Figure 6 b). In a next step the interior of the contour with its bounding walls is shaded in gray to be recognized by the algorithm as the unknown space. Figure 6 d) presents the grid-based costmap as the final output of the algorithm. The unknown space is still represented by the gray shaded cells. The walls are now classified as obstacles and shown in red. Moreover a predefined area around the obstacles is marked in pink to indicate the space occupied by the inflated obstacles.

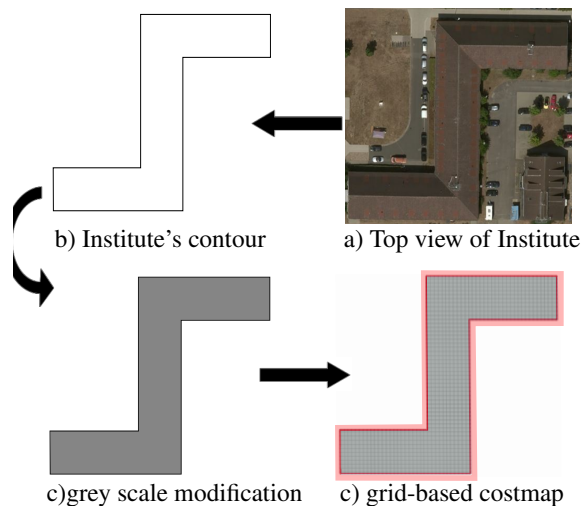


Figure 6: Image conversions

The exploration algorithm aims at cleaning out all the unknown space using the laser scanner data to update the information about the unknown cells and set them either to free or occupied. To do so the algorithm sets accessible goal poses by determining the frontiers of the free to the unknown space. For the purpose of frontier-based exploration a frontier is defined as a set of adjacent cells which share at least one edge with an unknown cell. In addition such a frontier must be wide enough to accommodate the exploring UAS. If both conditions are met the examined frontier is considered

accessible. In a next step a frontier goal is placed a predefined distance apart from the accessible frontier within the free space to preclude the collision with an unrevealed obstacle that might be hidden in the unknown space. All frontier goals are saved and updated every time new information have been added to the map. Among the current frontier goals the closest one to the robots position is submitted to the navigation algorithm which finally plans an obstacle-free path to the goal pose starting from the UAS pose. In case for a specified time no progress towards a goal pose is achieved, the currently pursued goal pose (frontier goal) will be put on a blacklist. Then, among the remaining frontier goals a new selection is performed by determining the frontier goal with the minimum euclidean distance to the UAS's position. Afterwards the determined closest frontier goal is passed to the navigation algorithm. As soon as all frontier goals are processed the quadrotor UAS will return to its initial point at the starting time of the exploration. Figure 7 presents the frontier goals identified by the exploration algorithm, the planned path towards the pursued frontier goal, the path traveled so far, the underlying costmap and the current position of the quadrotor UAS.

- frontier goal
- obstacle
- inflated obstacle
- unknown space
- travelled path
- pursued goal
- quadrotor
- path to pursued goal

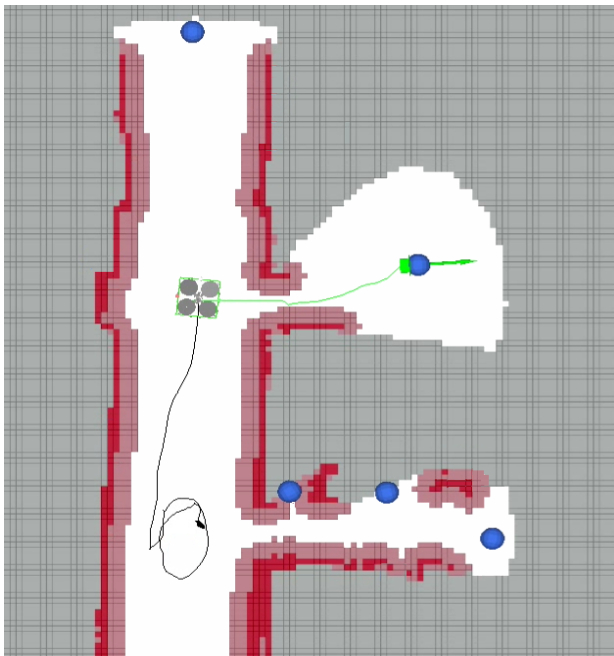


Figure 7: Exploration algorithm

3.3 Navigation

The navigation task is fulfilled by the combination of a global and a local planner. The global planner plans the global path from the UAS's current position towards the goal to pursue, avoiding the recognized obstacles, the area within a predefined distance around the obstacles and all cells of the grid map marked as unknown. The underlying path planning algorithm is part of the ROS navigation stack and computes the path using a grid-based map of the environment, which is identical to the one used by the exploration algorithm, and the well known Dijkstra algorithm. Furthermore to move the quadrotor UAS from its current position towards the goal along the global path linear and angular velocities have to be generated. This task is performed by the local planner which relies on the sensor measurements and a local grid-based map of the environment. The local map is centered around the quadrotor UAS and moves as the quadrotor UAS moves towards the goal. The map is centered around the UAS as the local planner just cares about obstacles within the area close to the quadrotor UAS. To determine the velocities two algorithms have been examined, whereas the Dynamic Window Approach has been favored over the Trajectory Rollout Planner as it samples from the set of achievable velocities for just one simulation step compared to sampling over the entire forward simulation period. Thus it is computational more efficient in generating the velocity commands [10]. Nevertheless both algorithms work in almost the same manner. The implemented Dynamic Window Approach [11] samples the UAS's three-dimensional control space $(\dot{x}, \dot{y}, \dot{\psi})$ and for each sampled angular or linear velocity the algorithm performs a forward simulation from the UAS's current state to predict what might happen if the sampled velocity is applied for a short period of time. The obtained trajectory is scored evaluating parameters such as proximity to an obstacle, proximity to the goal, proximity to the global path and the velocity. All trajectories that would result in a collision with an obstacle are excluded from the set of feasible trajectories. In the end, among the remaining trajectories, which correspond to a set of sampled velocities, the one with the highest score is chosen. Finally the velocities are used to derive the corresponding radio control commands, which are then submitted to the autopilot. Another feature of the Dynamic Window Approach is its ability to adapt the velocities to the environment. In the absence of obstacles and assuming the UAS can fly straight towards the goal, the algorithm will generate velocity commands that correspond to the UAS achievable maximum velocities. Whereas in an environment cluttered with obstacles appropriate linear and angular velocities are generated to avoid collisions by trying to maximize an objective function. This objective function includes a measure of progress towards the goal, the forward velocity of the UAS and the distance to the next obstacle on the path. Taking all three parameters into account, the UAS balances the objective to move fast towards the goal against the objective to

maneuver around the obstacles. The result of this behavior is a quite robust collision avoidance algorithm.

3.4 Mission control

Since the entire mission consists of several task an algorithm is deployed to initiate the single actions and switch between the several task regarding the feedback received from the autopilot or other algorithms, i.e. the exploration algorithm. The mission control program keeps track of the current state of the UAS, i.e. airborne or at ground, and switches between the available tasks for the current state in a predefined order. To initiate the switching between two consecutive tasks a feedback received from an algorithm or the autopilot is required, that the current task, i.e. the sensor initialization, the exploration or the landing procedure is accomplished. The exchange of information about the current progress of a task is handled using the ROS communication architecture. Since ROS is just running on the NUC board and thus it can be just used for the communication between the algorithms running on the NUC board as well another approach has been implemented to enable the communication between NUC board and the Pixhawk autopilot. This approach is described in the following section.

3.5 MAVLINK

To control the quadrotor UAS's motion by sending commands to the flight controller and to get the current state of the quadrotor i.e. for switching between different task, a communication link between the Pixhawk autopilot and the Robot Operating System must be established. As either of them the Pixhawk autopilot with its different firmwares and the Robot Operating System support the Micro Air Vehicle Communication Protocol (MAVLINK) it is implemented to establish the communication link between both. The MAVLINK protocol is an open source, header-only marshalling library which is already implemented in many modern autopilots to communicate between an UAS and its control station [12]. But likewise it can be used to enable the communication by wire between an onboard computer and an autopilot to exchange information or submit control commands. Currently the MAVLINK protocol can support 255 aircraft at once. The maximum message length is constrained to 263 bytes, whereas 8 bytes are reserved MAVLINK frame consisting of an message ID, component ID, system ID, the packet start sign, the payload length, the packet sequence and the checksum.

4 CONCLUSION

In this work an UAS system has been introduced which will be able to perform an autonomous indoor exploration. Therefore this system makes use of packages for navigation and path planning, which are already implemented and available for the Robot Operating System. This first version of the UAS system uses mainly a two-dimensional laser scanner for the environment perception and the exploration task.

5 FUTURE WORK

Though the designed system might be capable to sense and build a 3D map of the environment, it lacks the ability to plan a three-dimensional path. Furthermore it is not able to generate the appropriate commands to follow or track such a path. In addition to the abilities of planning and tracking an obstacle-free, three-dimensional path the sensor-based localization in a 3D environment might be considered in future to improve and enhance the autonomy of the UAS. Thus modifications and extensions of the current sensor configuration have to be made to sense the full 3D environment or performing a 3D-SLAM, i.e. by adding stereo vision camera. Furthermore an appropriate path planning algorithm and path tracking algorithm have to be implemented as well to enable the system to navigate in a cluttered 3D environment. The implementation and evaluation of the software and hardware is part of the ongoing studies and will be presented in future publications.

REFERENCES

- [1] G. Andrews S. Ahrens, D. Levine and J. P. How. Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments. In *IEEE International Conference on Robotics and Automation*, 2009.
- [2] R. He A. Bachrach, S. Prentice and N. Roy. Robust autonomous navigation in gps-denied environments. *Journal of Field Robotics (JFR 2011)*, 2011.
- [3] R. He A. Bachrach and N. Roy. Autonomous flight in unknown indoor environments. *International Journal of Micro Air Vehicles*, 2009.
- [4] R. He A. Bachrach and N. Roy. Autonomous flight in unstructured and unknown indoor environments. In *International Journal of Micro Air Vehicles*, 2009.
- [5] J. H. Kurz C. Eschmann and C. Boller. Cure modern - franco-german infrastructure inspection with unmanned air systems. In *International Micro Air Vehicle Conference and Flight Competition (IMAV2013)*, 2013.
- [6] A. Leber C.-H. Kuo, C.-M. Kuo and C. Boller. Vector thrust multi-rotor copter and its application for building inspection. In *International Micro Air Vehicle Conference and Flight Competition (IMAV2013)*, 2013.
- [7] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping. *IEEE Robotics and Automation Magazine*, 2006.
- [8] O. von Stryk S. Kohlbrecher, J. Meyer and U. Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, 2011.

- [9] B. Yamauchi. A frontier-based approach for autonomous exploration. In *International Symposium on Computational Intelligence, Robotics and Automation (CIRA)*, 1997.
- [10] E. Marder-Eppstein and E. Perko. Navigation stack - the base local planner. www.wiki.ros.org/base_local_planner. Accessed: 2014-05-09.
- [11] W. Burgard D. Fox and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 1997.
- [12] Lorenz Meier. The micro air vehicle communication protocol. www.qgroundcontrol.org/mavlink/start. Accessed: 2014-04-30.

Modular Quadrotor MAVs

Vishal Vaswani, Samuel Loh*, Benson Tan and Kalenjit Singh, Sutthiphong Srigrarom
University of Glasgow Singapore

ABSTRACT

Historically, helicopters with four rotors (quad-rotors) have been very uncommon, mainly due to the fact that most of the common payloads could be lifted using one or two rotors. However, the quad-rotor possesses some special characteristics that make it attractive. One would be the superior payload capacity. Two, is the simplicity of the control system: the absence of complex rotor mechanisms and just by independently adjusting the speed of each rotor it is possible to control both the attitude and the horizontal/vertical motion. This system is particularly suitable for small unmanned aerial vehicles (UAV), because it reduces the complex mechanism of the rotors (saving volume and weight) and simplifies the control algorithms required for stable, untethered flight. Although much progress has been made in the field of quad-rotor UAVs, it is still a great challenge to build a quad-rotor capable of fully autonomous flight. Before the decision of appropriate control algorithms, it is essential to have complete understanding of quad-rotor dynamics and equation of motion. This technical paper presents a detailed quad rotor model design as well as the detailed algorithm for the control system. An ANSYS flow simulation was carried out on the modular structure of the quad rotor. Different sensors were integrated with the control system to improve the stability and to reduce the pilot's workload. Its capabilities will be enhanced with semi-autonomous and autonomous functions that were implemented with the usage of GPS.

1. INTRODUCTION

A quad rotor is a rotorcraft equipped with four rotors laid up symmetrically around its centre. It is capable of hover flight, forward flight and vertical take-off and vertical landing. Therefore it is classified as a rotary-winged VTOL aircraft. Traditionally, the quad-rotor configuration is not of mainstream use in the aerospace industry, mainly because the usual payloads could be lifted using one or two rotors.

Much progress has been made in the field of quad-rotor UAVs; it is still a challenge to build a quad-rotor capable of stable, untethered flight with fully autonomous capabilities. Semi-autonomous capabilities will include safe low level flight and navigation in an urban environment. In order to enable this quad-rotor to fly autonomously, several controllers have been considered. It is expected that these controllers will be tested in the model before implementing them in the actual rotorcraft. To do so, it is necessary to implement the appropriate control algorithms and to have a suitable set of on-board sensors. As for the design of the

control algorithms, in order for it to be successful and stable, it is essential to have a complete understanding of quad-rotor flight dynamics.

2. QUAD ROTOR CONTROL

The configuration consists of two clockwise rotating rotors and two counter clockwise rotating rotors to compensate for the torque produced. The placement of the rotors can be seen in figure 1.

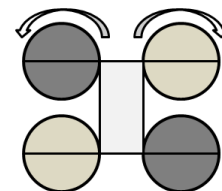


Figure 1: Control for H Frame Configuration

This unconventional design does not require a cyclic and collective pitch control mechanism of a conventional helicopter. Instead, each motor RPM is varied to replicate the exact controls of a conventional helicopter.

To increase its altitude, all four motors will increase to the same RPM. To pitch forward, the aft two motors are required to increase in RPM, while the forward two motors will have a slight decrease in RPM. Hence, a nose down or forward pitch attitude is produced about the x -axis and vice versa for pitch up attitude. This methodology of pitch can be used for roll about the y -axis as well. To yaw right, the clockwise rotating motors will increase in RPM and vice versa for yawing left. However, before designing the model of a quad rotor, quad rotor dynamics must be further explored.

3. QUAD ROTOR DYNAMICS

Quad rotor dynamics will include the equations of motion along all three axes. The force, moments and thrust equations are vital when designing a quad rotor. The general equations of motion can be written by considering various variables namely, force, moment of balance, moment of inertia and drag coefficients C_d . A fixed frame of body will be assumed to be at the center of gravity of the quad rotor with the z -axis pointing up. This body of axis is directly related to the three Euler angles ψ , θ , ϕ representing roll, pitch and yaw respectively. C_{d1} to C_{d6} are independent drag coefficients for the three body axes and the Euler angles.

*E-mail address: samuelohyk@gmail.com

$$\ddot{x} = \frac{F_{total}(\cos \psi \sin \theta \cos \phi + \sin \phi \sin \psi) - C_{d1}\dot{x}}{m} \quad (1)$$

$$\ddot{y} = \frac{F_{total}(\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) - C_{d2}\dot{y}}{m} \quad (2)$$

$$\ddot{z} = \frac{F_{total}(\cos \psi \cos \phi) - mg - C_{d3}\dot{z}}{m} \quad (3)$$

$$\ddot{\phi} = \frac{l(-F_1 + F_2 + F_3 - F_4) - C_{d4}\phi}{j_1} \quad (4)$$

$$\ddot{\theta} = \frac{l(-F_1 - F_2 + F_3 + F_4) - C_{d5}\theta}{j_2} \quad (5)$$

$$\ddot{\psi} = \frac{l(F_1 - F_2 + F_3 - F_4) - C_{d6}\psi}{j_3} \quad (6)$$

Assuming the initial condition of the model is in a low speed flight; drag is assumed to be negligible hence zero. Input equations are defined as:

$$U_1 = \frac{F_1 + F_2 + F_3 + F_4}{m} \quad (7)$$

$$U_2 = \frac{-F_1 + F_2 + F_3 - F_4}{m} \quad (8)$$

$$U_3 = \frac{-F_1 - F_2 + F_3 + F_4}{m} \quad (9)$$

$$U_4 = \frac{F_1 - F_2 + F_3 - F_4}{m} \quad (10)$$

U_1 represents total thrust acting on the body in the z-axis. U_2 and U_3 are the pitch and roll inputs respectively. U_4 is the yawing moment. Using actuator disc theory, the advance ratio and coefficient of thrust formulas, the thrust equation can be formulated.

$$\text{Advance ratio: } J = \frac{V}{nD} \quad (11)$$

where, D is the rotor diameter, V is the velocity and n is the rotor's rotational speed.

With the definition of coefficient of thrust:

$$T = C_T \rho n^2 D^4 \quad (12)$$

The values of advance ratio (J), coefficient of thrust (C_T) may be selected from the datasheet of propellers [2]. The center of gravity is assumed to be at the middle of the frame. A lower center of gravity will ensure a less sensitive angular acceleration. The tilting forces towards the center will reduce the pitching and rolling moments and vertical thrust. Hence, this results in an increase in static stability.

Although complex control mechanisms were eliminated, the quad rotor is a rotor winged craft and susceptible to the same aerodynamic effects as a conventional helicopter.

4. AERODYNAMIC EFFECTS

The quad rotor is modelled as a rotor winged aircraft. As such, it is subjected to various aerodynamic effects as a conventional helicopter would. It is however, possible to mitigate certain aerodynamic effects and at the same time to enhance flight performance.

Ground effect influences rotor operations close to the ground. There are two types of ground effect: In Ground Effect and Out of Ground Effect. In Ground Effect usually occurs less than one rotor diameter above the surface of the ground. As the induced airflow passing through the rotor disc is reduced by the surface friction, the lift vector increases. This allows a lower rotor blade angle for the same amount of lift, which reduces induced drag. In Ground effect also restricts the generation of blade tip vortices due to the downward and outward airflow making a larger portion of the blade produce lift.

Ground effect occurs on all types of surfaces. It is at its maximum effect in zero wind conditions over a firm and smooth surface (runway). Tall grass, uneven terrain, water surfaces will alter the airflow pattern which cause an increase in rotor tip vortices. The formation of tip vortices will be mitigated in the quad rotor's structural design.

When the rotorcraft moves through the air, the relative airflow through the rotor disc is different on the advancing side than the retreating side as seen in figure 2.

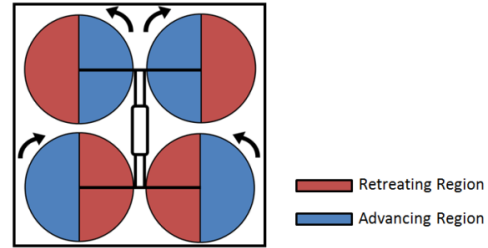


Figure 2: Advancing and Retreating Blade Regions [3]

The relative wind encountered by the advancing blade is increased by the forward speed of the helicopter. The relative wing speed acting on the retreating blade is reduced. Thus, the advancing blade produces more lift than the retreating blade. This phenomenon is known as Dissymmetry of Lift. This results in erratic pitching and rolling moments in flight. On a conventional helicopter, a horizontal hinge (flapping hinge) allows the individual rotor blades to flap as they rotate. [3] This unique system however, does not contain such a mechanical hinge. Instead, a complex control system will be implemented to reduce these unwanted moments.

5. STRUCTURE

A stable structure is required for testing aerodynamic effects associated with a rotor winged aircraft. In addition, a resilient and durable material will be a plus point as a

platform for a quad rotor system. In the market, there are numerous materials used in the design of a multi rotorcraft and three dimensional printing is increasingly popular due to its versatility. However, a revolutionary design that was manufactured by Safe Flight Copters LLC, in San Francisco, USA [4] focuses on the words ‘modular’ and ‘safety’. The structural design is best understood as an eggshell concept or also known as monocoque. While hollow in the interior, the design and material used can contribute a huge factor in improving its durability and strength. The manufacturing process is done by plastic injection moulding in which a plastic polymer was pressed onto a mould. Carbon fibre rods reinforce the frame. The plastic polymer is specifically known as polyethylene terephthalate (PET). It is one of the most common types of polyester. Two monomers, known as ethylene glycol and purified terephthalic acid are combined to form PET. PET has excellent wear resistance and has no centreline porosity. Typical properties of PET that are considered for this design are its tensile strength, tensile modulus, flexural strength, flexural modulus, dielectric constant, and dielectric strength [5]. A summary of the properties considered can be seen in table 2 of Appendix A.

This revolutionary design emphasises strongly on safety yet it is a modular design which is relatively rare in the market of multi rotorcraft. The modular concept essentially divides the quad rotor into parts. The parts consist of the lower fuselage & landing gear, upper fuselage, and two rotor modules as seen in figure 3.



Figure 3: Modular Design

The design feature requires no metallic linkages to connect the parts. Four rubber bands and puzzle joints secures the pieces together instead. Redundancy has been considered in this design in which the modules will not come apart while maintaining stable flight performance even if two of the four rubber bands have snapped. The plastic frame is designed to flex and disengage upon an impact or a crash. Hence there is a huge reduction in assembly time and maintenance cost.

A reduction of propeller tip vortices improves propeller efficiency. In other words, the generation of lift by the propeller will require less power for a specific throttle setting with shrouds installed. Rotor operations generate a suction pressure on the inlet. This results in additional lift. The total thrust will be contributed by the rotor and the shroud. The overall thrust with the installation of shrouds for a given

power input exceeds unshrouded rotors by about 30% as seen in figure 4 below.

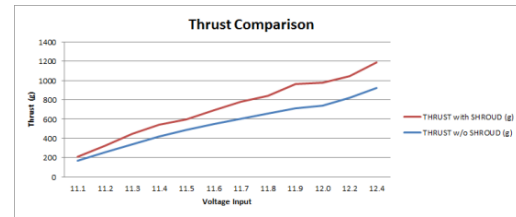


Figure 4: Motor thrust with and without Shroud

A slight increase in stability of hover flight was also observed. Propellers screen guards may also be installed as a first line of defence against foreign object debris that could stall a motor or damage the propeller in flight.

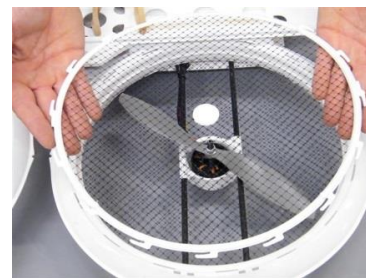


Figure 5: Propeller Screen Guards

The screen guards are made from polypropylene which is a thermoplastic polymer found in a variety of household products. The guards are resilient against foreign objects when tensioned by thermal appliances. However, installing a propeller screen guard also raised concerns of affecting the motor thrust. Using a tachometer and lining a reflective tape on the propeller, the individual motor RPM are as follows:

Throttle	Motor 1	Motor 2	Motor 3	Motor 4
50%	5700	5750	5725	5700
100%	11800	11850	11750	11850

Table 1: Screen guards installed

Throttle	Motor 1	Motor 2	Motor 3	Motor 4
50%	7050	7000	7020	7000
100%	13200	13100	13100	13200

Table 2: Screen guards removed

The simple tachometer test shows how the installation of screen guards reduces the thrust significantly by about 10%. This reduction of thrust is found to be acceptable performance losses for what is gained.

6. AVIONICS

The quad rotor is capable of both semi-autonomous flight (Pilot inputs/First Person View) and fully autonomous

capabilities. A general diagram of the avionics design on the quad rotor is shown in figure 6 below.

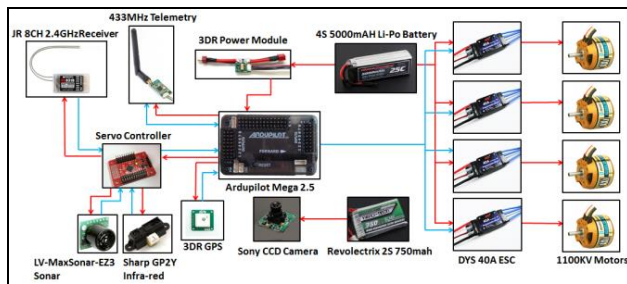


Figure 6: General Avionics Layout

Basic RC components consist of the brushless motor, propellers, electronic speed controller (ESC), lithium-polymer battery and an 8 channel transmitter/receiver. The list of components can be seen in table 5 of Appendix A. For the on board controller, an Atmel ATMEGA2560 microprocessor is used for processing information while an integrated barometer is used for high altitude sensing.

Based on the combination of weight of the components and weight of miscellaneous wires taken into consideration, the total weight of the rotor craft is estimated to be 1500 grams. The maximum thrust of each brushless motors produce with the chosen propeller is 1130 grams [6] with a size 10 propeller. A total of four motors will produce 4520 grams of thrust. The thrust to weight ratio would be 3:1. This amount of thrust is sufficient to propel the quad rotor and the capacity of the four cell lithium polymer (Li-Po) battery ensures flight endurance of 15 minutes after a flight test.

The propeller size was chosen due to the limitation of the propeller shroud design. A telemetry antenna operating at 433MHz ensures constant feedback of critical data such as current position to the ground station at all times.

During semi-autonomous flight, a servo controller has been integrated to regulate the radio signals from the receiver and signals of the implemented sensors. An ultrasonic sensor and infra-red sensors have been implemented in the system.

An ultrasonic sensor [7] enables altitude hold and autonomous take off/landing. The ultrasonic sensor functions in a similar principle to an aircraft's radar altimeter by taking reference to the datum. Atmospheric pressure is therefore negligible and the quad rotor will maintain set altitude regardless of sea level. The disadvantage of an ultrasonic sensor is it's only accurate on smooth surfaces at low altitude (15cm – 645cm). Carpet and irregular surfaces such as grass will interfere with the sensing capability and generate inaccurate distance readings. The servo controller essentially controls the amount of RPM the motors are required to generate to maintain altitude over time. In other words, the pilot is not required to control the throttle setting of the quad rotor. The graph below depicts the limitations of the sensor and also readings of the sensor. From the graph, it is observed that there is a 15 cm deadband. The height is

registered once the quad rotor has moved out of the dead band. The altitude hold can be manually set or programmed. The pre-set altitude used will be 175 cm. The time scale does not reflect the trigger time. This graph plot depicts the flight envelope and the accuracy of the ultrasonic sensor while maintaining altitude.

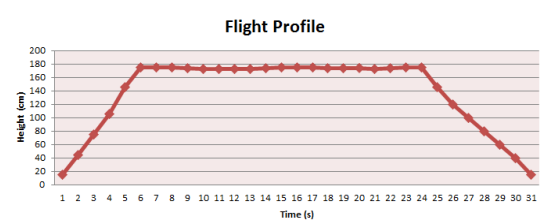


Figure 7: Ultrasonic Sensor readings

During the cruise phase, discrepancies in altitude readings can be observed. This is due to external disturbances from air-conditioning and pilot inputs.

An infra-red sensor has been implemented for obstacle & collision avoidance. The infra-red sensor's environmental perception is 25 centimeters with a quick trigger time of 50 milliseconds. An incremental corrective response has also been implemented. For every exponential increase of 5 centimeters, the servo controller will increase the respective motor RPM to accelerate the rate of correction. Below is a block diagram for the infra-red sensors operation.

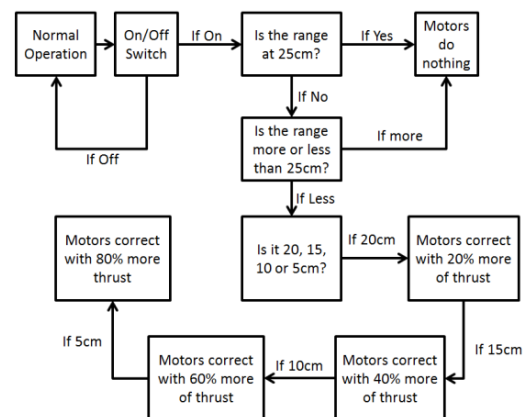


Figure 8: Obstacle & Avoidance System

The infra-red system and ultrasonic system can be removed and attached to interact with any Arduino based flight controller in the market. This is to prevent any discrepancies when the firmware or program in the flight controller has been updated. If required, the user may choose to remove the implemented sensors and install other types of sensors with the servo controller to enhance the capabilities of the quad rotor.

A charge-coupled-device (CCD) phase alternating lines (PAL) camera is used to capture images and provide video

feedback to the ground station. An independent two cell lithium polymer battery generates enough power for the camera to last up to a few hours, which is more than sufficient. The camera may be installed at the nose of the quad rotor which contains a servo to pan the view vertically. If preferred, an aft mount can also be used to install a camera for third person view.

Basic RC components and pilot inputs are insufficient for minute corrections. A complex control system must be implemented for auto stabilization along all 3 axes.

For any radio controlled aircraft, safety is paramount. Regulations require the aircraft to have failsafe on board in the event of a loss in data link. The controller has two types of failsafe. The first failsafe is the arm/disarm feature. After switching on the transmitter and connecting the battery, an arm sequence is required to send current to the brushless motors. Without this arming sequence, the motors will not spin even if the pilot decides to increase the throttle stick. The arming of the controller is achieved by pushing the yaw stick to the left for five seconds. This command will be input into the controller and the motors will be armed for flight. For landing, the user is required to push the yaw stick to the right for three seconds to disarm the motors. This failsafe function runs on the secondary PPM encoder processor.

The encoder also detects the loss of RC signal in the RC receiver and will initiate a user defined set of autonomous commands. These commands can range to loiter about the last known position, return to its initial starting point (Home) or perform an autonomous landing. This failsafe function will require calibration to the radio transmitter as well. The user needs to lower the throttle PPM value than the lowest stick position during the first binding of transmitter and receiver. This setting will tell the controller that this is not a user induced action but rather a loss of data link. Therefore, the controller will not initiate a user defined command when the pilot reduced the throttle stick to zero.

7. CONTROL

For auto stabilization along 3 axes, an inertial measurement unit (IMU) will be implemented into the system based on direction-cosine-matrix (DCM) [8]. The theory is to use gyro information as the primary link between the reference frames while using global positioning system (GPS) and accelerometer information to compensate for gyro drift.

The orientation of the aircraft can be described by three consecutive rotations, whose order is important. The angular rotations are called the Euler angles. The orientation of the body frame with respect to the fixed earth frame can be determined from equations (13) to equations (17) and can be seen in figure 15.

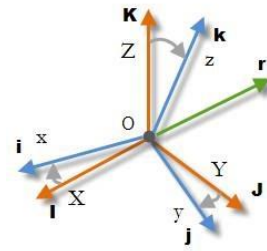


Figure 9: Orientation of Body to Earth coordinates

The transformation to a fixed body frame of reference from an inertial frame of reference (Earth) must be performed. Essentially, the transformation matrix requires three rotations about the z axis, y axis and finally the x axis.

$$R_z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

$$R_y = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (14)$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (15)$$

$$R = R_z R_y R_x \quad (16)$$

$$R = \begin{bmatrix} c\psi.c\theta & c\psi.s\theta.s\phi - c\phi.s\psi & c\psi.s\theta.c\phi + s\phi.s\psi \\ s\psi.c\theta & s\psi.s\theta.s\phi + c\phi.c\psi & s\psi.s\theta.c\phi - s\phi.c\psi \\ -s\theta & s\phi.c\theta & c\phi.c\theta \end{bmatrix} \quad (17)$$

where, c = cosine and s = sine.

The R matrix above contains all the information needed to express the orientation of the plane with respect to the ground. This matrix is also called the direct-cosine-matrix because each cosine angle is the angle between an axis of the aircraft and an axis on the ground. Electronic rate gyros rotate with the aircraft, producing signals proportional to the rotation rate. Since rotations do not commute and the sequence of rotations is important, the integration of gyro rate signals to obtain angles will not work. The kinematics of rotations must be looked into.

Rate of change of a rotating vector:

$$\frac{dr(t)}{dt} = w(r) \times r(t) \quad (18)$$

$w(r)$ is the rotation rate vector.

By numerically integrating a cross product to track the rotating vector,

$$r(t) = r(0) + \int_0^t d\theta(t) \times r(t) \quad (19)$$

$$d\theta(t) = w(t) dt \quad (20)$$

$r(0)$ = starting value of the vector

$\int_0^t d\theta(t) \times r(t)$ = change in the vector

Equations (19) and (20) will be applied to the rolls and columns of Matrix R as rotating vectors. Ideally, the axes of the aircraft in the earth frame of reference will be tracked. Since the gyro measurements are made in the aircraft frame of reference, the symmetry in the rotation must be recognized. The earth axes can be tracked in the plane frame by flipping the sign of the gyro signals.

$$r_{earth}(t) = r_{earth}(0) + \int_0^t r_{earth}(t) d\theta(t) \quad (21)$$

$$d\theta(t) = w(t) dt \quad (22)$$

r_{earth} = one of the earth axis

The vectors in the above equations are the rows of Matrix R in equation (1.21). To implement this, the differential form of the above equations is:

$$r_{earth}(t + dt) = r_{earth}(0) + r_{earth}(t) \times d\theta(t) \quad (23)$$

Lastly, the correction rotation rate must be added that comes out of the proportional + integral drift compensation feedback controller, to the measurements made by the gyros to produce the best estimation of the true rotation rate.

$$w(t) = w_{gyro}(t) + w_{correction}(t) \quad (24)$$

$w_{gyro}(t)$ = three axes gyro measurements

$w_{correction}(t)$ = gyro correction

Even though the gyros function well, the uncorrected offset of a few degrees per second must be cancelled. The rotation error vector can be fed back through a proportional plus integral (PI) controller to produce a rotation rate adjustment for the gyros. The derivative term may be used to improve stability.

Accelerometers are used for roll and pitch drift correction because they contain zero drift. Centrifugal acceleration will also be accounted for as the cross product of the gyro vector and velocity vector.

$$A_{centrifugal} = w_{gyro} \times V \quad (25)$$

$$V = \begin{bmatrix} \text{velocity} \\ 0 \\ 0 \end{bmatrix} \quad (26)$$

To estimate gravity that is adjusted for centrifugal acceleration, the centrifugal acceleration estimate must be added. Hence:

$$g_{reference} = \text{Accelerometer} + w_{gyro} \times V \quad (27)$$

$$\text{Accelerometer} = \begin{bmatrix} \text{Accelerometer}_x \\ \text{Accelerometer}_y \\ \text{Accelerometer}_z \end{bmatrix} \quad (28)$$

In addition to the reference measurement of gravity, estimation must be done based on the DCM. The roll and

pitch rotational correction vector in the body frame of reference is computed by taking the cross product of the Z row of the DCM with the normalized gravity reference vector (Equation 27).

$$\text{RollPitchCorrectionPlan} = \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix} \times g_{reference} \quad (29)$$

Gyros tend to saturate during tight turns. Actual acceleration might exceed the range of the accelerometer. This can be avoided by adding gyro terms in the feedback controller to limit the turning rate.

In a nutshell, the laws of DCM depict the controller to use gyro information to integrate the time rate of change of the direction cosines. Speed and gyro information are used to adjust accelerometer information for centrifugal effects. 3 axis accelerometers information are used to cancel roll and pitch drift. Lastly, GPS information will be utilised to cancel yaw drift.

A proportional-integral-derivative (PID) controller is a control closed loop feedback system. The PID controller calculates the error value as a difference between the measured value and a desired set point. The comparison is then multiplied times Rate P. The PID will ensure the commanded rate matches the actual rate from the gyro. Values will then be subsequently converted into Pulse Width Modulation (PWM) and set to the motors via the ESCs.

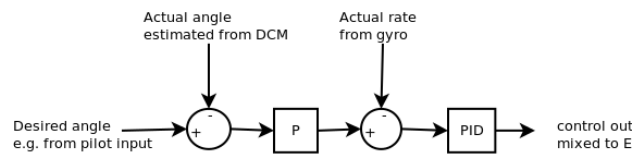


Figure 10: PID Control Diagram

Increasing Proportional gain will increase the rotation rate as desired to correct any errors. A greater value will result in a faster attempt to achieve the desired attitude. However, an exceptionally high value will result in the correction of pilot's input almost immediately which is considered undesirable. Integral term can be used to account for weak motors or strong external forces. Again, if the value is too high, oscillations will be observed. Derivative term in the system will predict behaviour and improve settling time and overall stability. The final results for the PID values are:

$$\text{Rate_P} = 0.170$$

$$\text{Rate_I} = 0.650$$

$$\text{Rate_D} = 0.008$$

PID controllers however, are insufficient for fully autonomous flight. An additional component must be implemented.

8. MAN-MACHINE-INTERFACE

A first person view (FPV) camera has been implemented to provide real time video feedback to the ground station or pilot. A split screen will be required to display video and air data. But this set up can be drastically improved by using the video display to replace the artificial horizon as a heads up display (HUD) through graphic user interface (GUI). This combination of GUI and FPV is as shown in figure 11 below.

Critical air data would include an artificial horizon, airspeed, altitude, heading, mode of flight, battery voltage and capacity. This integration improves pilot's situational awareness and reduces the pilot's workload during semi-autonomous flight. In addition, the video can be recorded at any point of time to be used for analysis if needed.



Figure 11: Heads up Display with FPV Camera

9. AUTONOMOUS FLIGHT

To achieve outdoors autonomous flight, a GPS module can be implemented with the controller.



Figure 12: 3DR GPS Module

Most GPS receivers are using pseudorange data available on the GPS L1 channel of 1675.42MHz. Optimally, the receiver receives satellite based augmentation system (SBAS), differential global positioning system (DGPS) corrections which give meter precision. However, an advanced and expensive method can provide centimeters or even millimeters of precision. These GPS receivers uses the previous method as mentioned and at the same time with carrier phase smoothing, carrier phase corrections or Real

Time Kinematic (RTK) as well as L2 channel semi-codeless tracking of 1227.60MHz. RTK uses pseudorange to compute position and at the same time, Doppler measurements and carrier phase smoothing. The base station collects these data and transmits them to the receiver on the quad rotor. Since the base is on a known coordinate, computations can go up to millimeters accuracy.

In addition, these account for real time local ionospheric corrections which provide pin point accuracy.

GPS may be accurate for the outdoors autonomous flight, but a problem known as multipathing causes GPS position errors [9]. This problem is difficult to detect and compensate but through understanding, they can be avoided. Multipathing results when the direct path to the receiver is reflected by an object. The reflecting surface may be from buildings, mountains, trees and even the human body. A radio reflector at 1.6GHz can also cause multipathing. When the path is blocked, the radio signals have to travel further to get to the receiver than they should. The GPS will miscalculate its position because the signals have travelled further than a direct line of sight as in figure 13 below.

Ionospheric perturbations and magnetic storms can cause signal delays as well. This is partially compensated using SBAS and L1/L2 decoding receivers. A general rule of thumb for autonomous flight using GPS would be to fly with clear sky view and no obstructions.

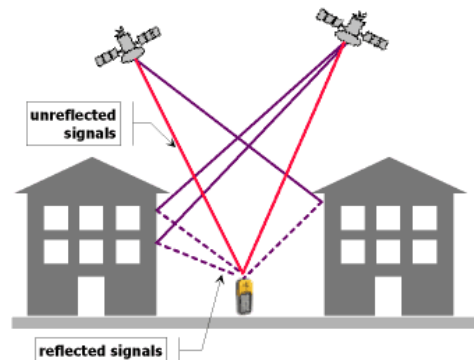


Figure 13: GPS Multipathing

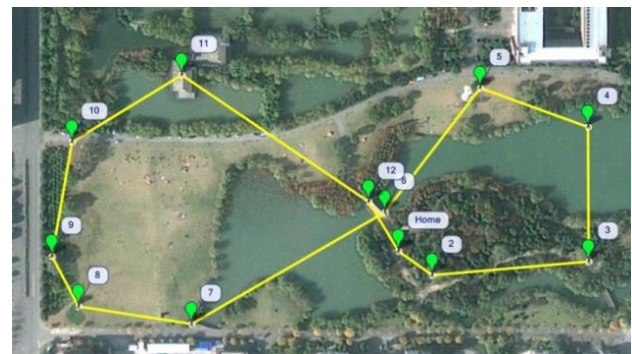


Figure 14: User Interface for GPS way points

A utility tool known as Mission Planner can be used as an interface in conjunction with the Ardupilot Mega 2.5

controller. This tool interacts with the controller which enables the user to insert or select GPS coordinates, and set them as way points for the quad rotor to fly.

A GPS failsafe has also been incorporated. This GPS failsafe event will occur if the GPS position “glitches” while the quad rotor is in autonomous GPS mode. A projected path and GPS reported path will be relayed to the controller. The projected position will be based on the previous GPS position and velocity. GPS position “glitches” are detected by comparing each new position update (every three seconds) received from the GPS with a position projected out from the previous position and velocity. A radius around a way point can be set to allow positions within the area to be accepted. The radius can range from 5 meters to 10 meters. Once within the radius, the controller “accepts” the new position and the cycle repeats. If the GPS position updates outside the boundary of the projected and acceptable radius, it triggers a user defined GPS failsafe command after five seconds. Manual radio control via a pilot can also be utilised in such a situation.

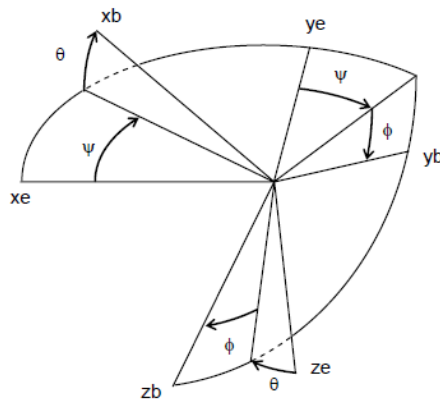


Figure 15: Body to Earth frame of reference

10. CONCLUSION

The aim of this project was to have a fully functional quad rotor capable of semi-autonomous control and autonomous flight. During test runs, we achieved perfect stable flight and autonomous functions while exhibiting visible corrections from the controller. The physical structure is successful both in terms of weight, strength and durability. The final weight was measured to be 1800 grams. Fully functional software was implemented for the control of the four motors on a commercially available microcomputer and obtained excellent results with an oscilloscope set up to compare and read operator inputs & PWM inputs. The autonomous aspect of the project was also successful. Fully autonomous functions were observed without the intervention of the pilot aside from monitoring of flight. Future developments would include fine tuning of the controller and outfitting with LiDAR (Light Detection And Ranging) [10] for precise autonomous indoors navigation.

Structural enhancements such as a retractable landing gear and aerodynamic body can also be looked into.

References

- [1] Leishman, G., *Basic Helicopter Aerodynamics*, Cambridge University Press, 2003.
- [2] Norris, D. *Build your own Quadcopter*, McGraw-Hill, 2014.
- [3] Federal Aviation Administration, *Helicopter Flying Handbook*, Chapter 5 – Helicopter Components, Sections, and Systems. 2012.
- [4] Ananda G., University of Illinois, Department of Aerospace Engineering, Propeller Database 2014: [ONLINE] Available at: <http://aerospace.illinois.edu/m-selig/props/propDB.html>
- [5] Federal Aviation Administration, *Rotorcraft Flying Handbook*, Chapter 5-4 Main Rotor System. 2000.
- [6] Safe Flight Copters LLC, San Francisco, California, United States of America: [ONLINE] Available at: <http://safeflightcopters.com/SFC4410>
- [7] Plastic Properties INC, Bessemer City, North Carolina, United States of America. PET Properties [ONLINE] Available at: <http://www.plastic-products.com/part12.htm>
- [8] Hobbyking, Turnigy D2836/8 1100KV, Brushless Motor datasheet, 2014. [ONLINE] Available at: http://www.hobbyking.com/hobbyking/store/uh_viewitem.asp?idproduct=18969
- [9] Maxbotix LV-Maxsonar EZ-3 Ultrasonic Sensor Datasheet, 2012. [ONLINE] Available at: http://www.maxbotix.com/documents/MB1030_Datasheet.pdf
- [10] Direct Cosine Matrix, Starlino Electronics, 2011. [ONLINE] Available at: http://www.starlino.com/dcm_tutorial.html
- [11] Joe Mehaffey, GPS Multipathing, 2013. [ONLINE] Available at: <http://gpsinformation.net/multipath.htm>
- [12] National Oceanic and Atmospheric Administration, Light Detection And Ranging (LiDAR), 2013. [ONLINE] Available at: <http://oceanservice.noaa.gov/facts/lidar.html>

APPENDIX A: DATA

Components	Description
Controller	Ardupilot Mega 2.5
Electronic Speed Controller	Dys 40A Programmable ESC
Brushless Motor	Turnigy 1100KV
Propeller	APC 10 by 4.7
Global Positioning System	3DR GPS Module
Infra-red Sensor	Sharp GP-2Y
Ultrasonic Sensor	LV-Maxsonar-EZ3
Battery	14.8 4S 5000mAh Li-Po
Camera	CCD Camera
Transmitter/Receiver	JR 8 channel TX/RX
Wireless Data link	3DR 433MHz Radio Telemetry

Table 3: Summary of Components

Lisa-S 2.8g autopilot for GPS-based flight of MAVs

B.D.W. Remes,* P. Esden-Tempski,† F. van Tienen, E. Smeur, C. De Wagter, and G.C.H.E. de Croon
Delft University of Technology, Kluyverweg 1, Delft, the Netherlands

ABSTRACT

Recent advances in sensor miniaturization have enabled the development of a miniaturized fully functional autopilot. In this article, the open hardware and open software **Paparazzi-UAV** **Lisa-S** micro autopilot is presented, weighing only 2.8 grams and measuring 2 by 2 centimeters. It incorporates 2-way telemetry with a ground station, a Spektrum DMS2/DSMX remote control, an IMU (Inertial Measurement Unit), a Magnetometer, a GPS module and a 32-bit 72 MHz processor for stabilization and navigation. As an illustration, the **Lisa-S** has been mounted on various miniature multi-rotors. Mounting it on a Ladybird quadrotor leads to a 8.5 by 8.5 cm system weighing 27 grams (excluding battery). Flight results are shown of the Ladybird performing autonomous GPS-based navigation missions outdoors.

1 INTRODUCTION

Micro Air Vehicles (MAVs) can serve a wide range of different missions, requiring a certain level of autonomous flight. The core element making this possible is the *autopilot*, which controls the vehicle on the basis of incoming (sensory) information. A crucial capability for many current practical applications, such as land survey or exploration of outdoor environments, is GPS-based flight [1, 2].

While most of the current applications and research involve MAVs of a mass ranging from 500 grams to 1 kg (cf. [3]), there is an increasing focus on much smaller MAVs in the order of tens of grams [4, 5, 6, 7]. While such MAVs have a high potential to be widely used thanks to their small mass and size, they typically lack a functional miniature autopilot with onboard autonomous GPS navigation. An example of a 16 gram MAV that does have this capability is the 'Black Hornet' [8]. This (expensive) MAV is made for military use, and is equipped with a fully integrated closed hardware and closed source autopilot. In order to make significant progress both in science and for civil applications, it is highly desirable to have a low-cost, powerful, and generic miniature autopilot that allows modifications to both hardware and software.

In this technical paper, we present the (currently) smallest open source and open hardware autopilot in the world,

which can be used on very small and light-weight MAVs. The autopilot design is based on the open source **Paparazzi-UAV** [9, 10, 11, 12, 13] autopilot framework.

Paparazzi-UAV has officially supported at least 13 different autopilot hardware designs over the past 10 years. When integrating these autopilot boards as well as other open-source autopilot designs [13, 14] into a UAV, the realization was made that a significant amount of the weight, size and complexity came from the need to attach additional sensors, data links, GPS receivers, remote control links and motor controllers to the main autopilot board. The total system hereby very often became much larger than the autopilot board itself.

With the design and production of the miniature autopilot **Lisa-S** all those issues were addressed. This fully functional autopilot weighs only 2.8 grams and measures $2 \times 2 \times 0.5$ cm. The design of the **Lisa-S** autopilot was done in close cooperation between the MAVlab¹ of the TUDelft and 1 Bit Squared², commissioned by the MAVlab.

In the remainder of this article, we first describe the **Lisa-S** main board and its development (Section 2). Subsequently, we delve further into the communication module, the SuperbitRF (Section 3). The use of the **Lisa-S** is illustrated by means of installing it on nano quad rotors, of which flight test results are shown (Section 4). Finally, we conclude in Section 5.

2 LISA-S

2.1 Hardware description

The **Lisa-S** autopilot incorporates all sensors and computer power to fly an unstable platform like a multirotor, autonomously towards defined GPS coordinates. Everything is integrated in a 20 by 20 by 5 millimeter package with a weight of only 2.8 grams. The computation power comes from a 72MHz 32bit ARM Cortex-M3 MCU with 64KB RAM and 512KB Flash. The Inertial Measurement Unit (IMU) sensors consist of an integrated 3-axis Gyroscope and 3-axis Accelerometer sensor from InvenSense MPU-6000, the 3-axis magnetometer is the Honeywell HMC5883L and the barometer (altimeter) is the Measurement Specialties MS5611.

For GPS waypoint navigation a U-Blox Max-7Q module is integrated in the design. As a backup power source for the GPS a 0.011 Farad PAS3225P3R3113³ supercapacitor is used. When swapping the battery of your MAV this capacitor

*Email address: b.d.w.remes@tudelft.nl

†Email address: piotr@1bitsquared.com

¹<http://www.mavlab.lr.tudelft.nl/>

²<http://1bitsquared.com/>

³Digikey 587-3260-6-ND

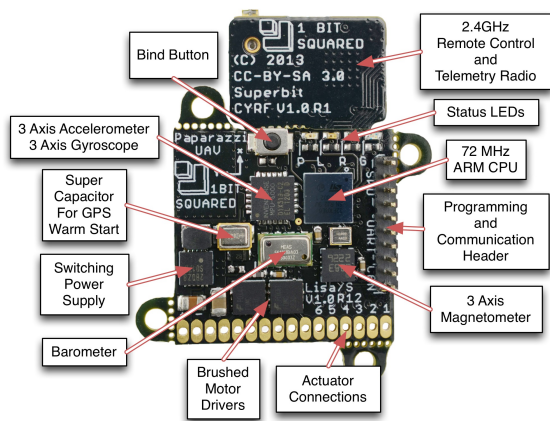


Figure 1: Lisa-S V1.0 R12 top: showing processor, IMU sensors, switching voltage regulator, GPS backup power super capacitor and dual MOSFET brushed motor drivers

keeps the U-Blox in backup battery mode for ~20 minutes and a cold start is avoided when swapping batteries.

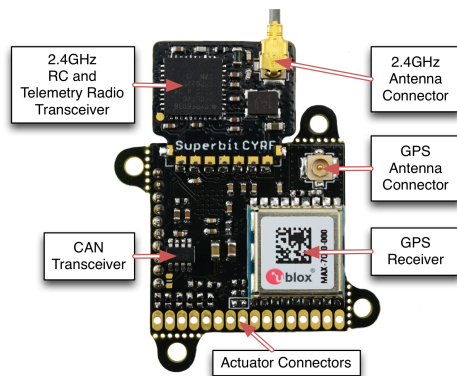


Figure 2: Lisa-S V1.0 R12 bottom: showing Ublox GPS module, CAN (Control Area Network) transceiver and Superbit CYRF radio module

To make the Lisa-S autopilot versatile and usable on a variety of platforms as for example: multicopters, fixed wing airplanes, helicopters or transitioning VTOL systems as well as ornithopters, great care has been given to the functions of the autopilot hardware. The Lisa-S has 6 PWM (servo) outputs. As an extra feature 4 of these can be used to directly drive DC motors thanks to Fairchild Semiconductor FDMC8030 MOSFET switches which are connected to 4 of the 6 PWM output channels. A Superbit CYRF RC and telemetry module can be added by simply soldering it to the respective pads on the autopilot. The autopilot is equipped with a Texas Instruments LM3668SD-3034 switch-

ing buck/boost converter, allowing a wide range of input voltages, making it perfect for operation from a 1S LiPO battery. To make the Lisa-S a powerful developers and researcher's tool, a UART port, a Maxim MAX3051EKA CAN interface and SWD programming/debugging interface are provided. Finally, the Lisa-S has a Bind/Boot tactile switch on the top side of the board. Hardware developments on the Lisa-S can be found on the [Paparazzi-UAV Lisa-S wiki page](http://wiki.paparazziuav.org/wiki/Lisa-S)⁴

2.2 Previous version

The initial design of the Lisa-S autopilot was initiated by the MAVlab of the TUDelft to make the Walkera Genius CP V2⁵ nano helicopter fly autonomous. This led to the 20 by 20 millimeter size restriction and constrained weight limit. Most of the design is similar to the final version, but there are some differences.



Figure 3: LISA/S V0.1 bottom: showing Ublox GPS module

Because a helicopter has 2 motors, only 2 (instead of 4) of the 6 PWM outputs were equipped with MOSFET DC motor drivers. In the first design, the GPS had no supercapacitor that is now serving as a backup power source while the main battery is being replaced. Every time the Lisa-S was powered up the U-Blox GPS module had to reacquire all the satellite information from scratch, resulting in a very lengthy reboot cycle, known as a cold start. The addition of the supercapacitor allows the GPS to reuse information acquired previously to shorten the time significantly from several minutes to seconds.

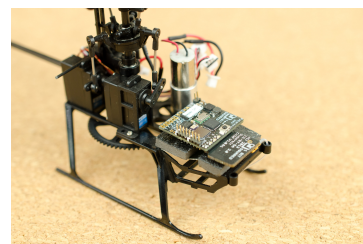


Figure 4: LISA/S V0.1 top mounted on a Walkera Genius CP V2 nano helicopter

No mounting holes were included in the initial design so

⁴<http://wiki.paparazziuav.org/wiki/Lisa-S>

⁵<http://www.walkera.com/en/showgoods.php?id=492>

mounting was done with a piece of double sided sticky foam tape. This had the disadvantage that a calibration step was needed before every flight.

3 SUPERBITRF

3.1 Hardware description

The SuperbitRF hardware is comprised of three main parts. The first part is the Superbit CYRF module, mounted on both the autopilot and the USBRF dongle. It is a miniature carrier board of 0.6 grams for the Cypress Semiconductor CYRF6936 2.4GHz radio transceiver chip⁶. To save weight it has no dedicated MCU, but instead it uses the 72MHz 32bit ARM Cortex-M3 MCU of the autopilot to implement the SuperbitRF protocol. During the design process the SuperbitRF module was designed as a separate part, with the intention of extending the capabilities of the communication module in the future. Eg.: adding an amplifier for extended range or changing the transceiver of a higher throughput radio transceiver supporting an integrated video stream.

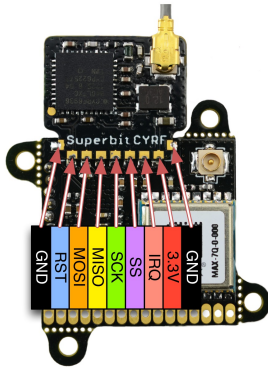


Figure 5: Lisa-S with Superbit CYRF module

The second part is the SuperbitRF USBRF dongle, designed for the ground control station computer. It consists of a USB enabled 72MHz 32bit ARM Cortex-M3 MCU connected to the previously described SuperbitRF module. The SuperbitRF protocol and USB stacks are implemented on the ARM microcontroller, and it even supports other protocols.

The third part is a standard off-the-shelf Spektrum handheld RC transmitter, transmitting DSM2 or DSMX protocols. This transmitter is used for flying the MAV manually and to take over control if required for safety.

3.2 Protocol

The main focus of the SuperbitRF is maintaining a stable RC connection with an original Spektrum⁷ transmitter, while adding the support of a datalink connection with the USBRF

⁶<http://www.cypress.com/?docID=48819>

⁷<http://www.spektrumrc.com/>

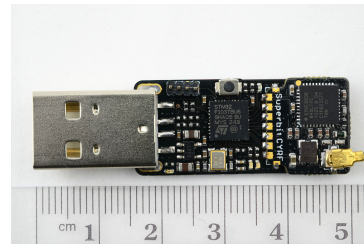


Figure 6: Superbit USBRF dongle

dongle. This is done by extending the Spektrum DSM2 and DSMX protocols[15] and thereby creating the new SuperbitRF protocol.

Reverse engineering and the work of Deviation⁸ enabled a full understanding of the DSM2 and DSMX protocols. The original Spektrum RC protocol works by transmitting a maximum of 7 RC channels every 22 or 11 milliseconds on a certain RF channel in the 2.4GHz frequency band. After transmitting on this first RF channel it transmits the same 7 RC channels for redundancy on another RF channel, called frequency hopping. As a way of dealing with the interference of other Spektrum transmitters, the original protocol uses several tricks, one of which is sending two bytes of the transmitter identifier in every RC packet.

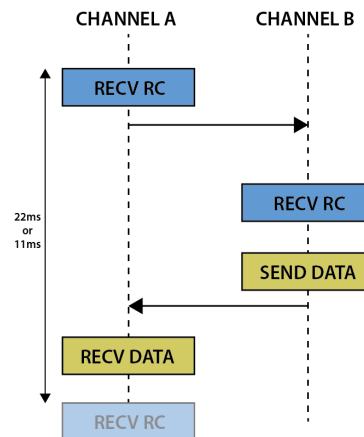


Figure 7: SuperbitRF protocol

After analysing the original Spektrum protocol a possibility was found to extend it with a datalink connection. Various ways of extending the protocol were considered and focused mainly on the 22ms DSM2 transmitters. By focussing mainly on the 22ms DSM2 transmitters the original RF redundancy packet was kept and the datalink was implemented without timing issues. The implementation of the SuperbitRF datalink is done by sending telemetry to the USBRF dongle on the redundancy channel right after the RC packet is re-

⁸<http://www.deviationtx.com/>

ceived. It then switches to the first channel to receive an up-link packet and another RC packet on the first channel.

By implementing the SuperbitRF extensions protocol as described, a safe RC connection was maintained even without the USBRF dongle for the datalink connection. To prevent the SuperbitRF datalink from interfering with the original RC connection, a different transmitter identifier was used while sending the telemetry and uplink packets.

4 NANO QUADS

4.1 Hardware description

To test the performance of the Lisa-S autopilot, it is used as a stabilization and navigation computer for unstable MAVs: namely micro quadrotors. The first quadrotor built with the Lisa-S V0.1 was the 'Chouchou', which is an 5.7 by 5.7 cm, 26.5 gram (including 260mAh battery) brushless quadrotor. Micro SiLabs MCU based Supermicro 3.5A ESCs are connected to 4 Lisa-S PWM outputs. The ESCs are flashed with the open source BLHeli firmware⁹ for better performance and the AP01 Micro Brushless 1100KV motors are used to drive the propellers from the Silverlit X-Twin R/C Bi-Wing airplane¹⁰.

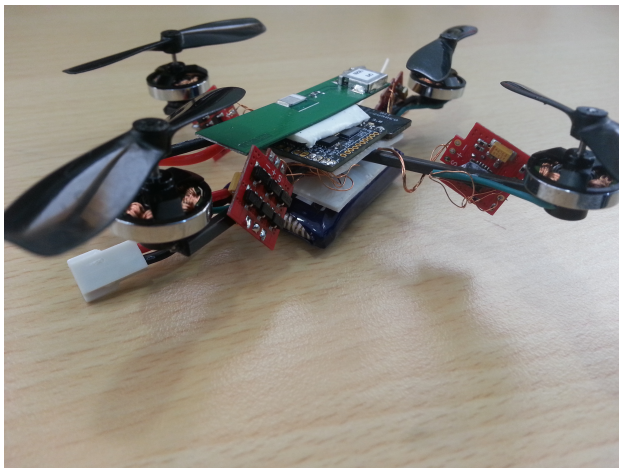


Figure 8: ChouChou 5.7x5.7cm brushless quad with Lisa-S and Taoglas PCB GPS antenna

The mounting holes of the Lisa-S V1.0 R12 are designed to fit on the 8.5 by 8.5 cm Walkera Ladybird quadrotor¹¹. To achieve this, the mounting holes are placed on the side of the Lisa-S resulting in a 13 degree rotational offset around the yaw axis. This can be easily adjusted in the Paparazzi-UAV software, by adding a "body to IMU" parameter. The mounting holes are made optional by the addition of "mouse

⁹<http://www.rcgroups.com/forums/showpost.php?p=25972443&postcount=3>

¹⁰http://http://www.silverlit-flyingclub.com/xtwin_classic.htm

¹¹<http://www.walkera.com/en/showgoods.php?id=467>

bite" breakoff points, making the tabs containing the mounting holes removable. If necessary the size of the main board can be reduced to the original dimensions of the Lisa-S V0.1 board design. To facilitate the conversion of the Ladybird to an autonomous system, an additional PCB (Printed Circuit Board) adapter is designed to fit the original Ladybird motor connectors. The converted RTF autonomous Walkera ladybird weighs 27g without battery. As an example, one can add a 350mAh battery weighing 8 grams, resulting in a weight of 35 grams.



Figure 9: Ladybird 8.5 by 8.5 cm with Lisa-S and the Taoglas Ceramic GPS antenna

To achieve good autonomous flight performance a high sensitivity GPS antenna is required. With weight and size limitations in mind, only two candidates were found suitable. The Taoglas 1575 MHz PCB Active Loop Module ALA.01.07.0095A¹² of 1.3 grams and measuring $45 \times 10 \times 2.3$ millimeters. And the Taoglas AP.10F.07.0039B 1575.42MHZ Ceramic Active LNA circuitry¹³ of 3 grams and measuring $10 \times 10 \times 5.7$ millimeters.

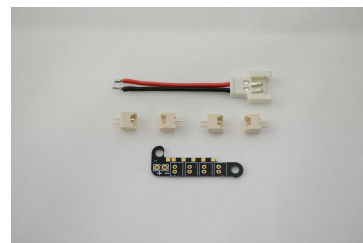


Figure 10: Ladybird connector PCB for Lisa-S

4.2 Flight test

Flight testing was performed with the Chouchou and the Ladybird quadrotors. The Chouchou quadrotor was first

¹²http://www.taoglas.com/images/product_images/original_images/ALA.01.07.0095A.pdf

¹³http://http://www.taoglas.com/images/product_images/original_images/AP.10E.07.0039B.pdf

tested indoors and flown manually in stabilization mode. Resulting in a stable indoor flight, which proves the **Paparazzi-UAV** software was running fast enough to stabilize the unstable platform. Despite several outdoor flight attempts with the Chouchou, no successful autonomous GPS supported navigation flights were performed. The source of the issue can be found in the lack of an interference free spot for the Taoglas PCB GPS antenna to be placed on the tiny Chouchou airframe. Future attempts will be made with the Taoglas ceramic GPS antenna.

The Ladybird quadrotor was first flown successfully indoors manually in stabilization mode. Several autonomous GPS supported navigation flights were performed which included takeoff, waypoint flying and landing (figure 11). The MAXTENA MIA GPS antenna was fitted on top of the MCU. This location is the spot with the least interference for the GPS, resulting in a position accuracy of less than 3 meters. Figure 11 shows the trajectory of the Ladybird during one of the flights.

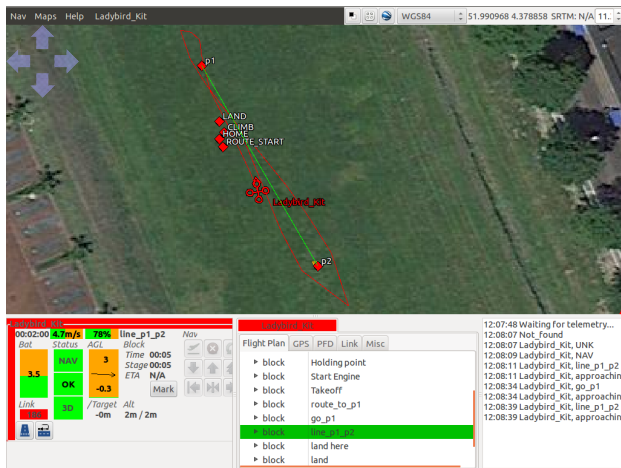


Figure 11: **Paparazzi-UAV** ground station during an autonomous flight of a Lisa-S equipped Ladybird

4.3 Lisa-S Technical Specifications

For full **Paparazzi-UAV** Lisa-S system specifications please refer to Table 1.

5 CONCLUSION

When it comes to autopilot size, one should not consider just the central processing board but the entire system. A miniature 2 by 2 centimeter autopilot board was developed called **Lisa-S** which includes all the required components for both manual remotely controlled as well as fully autonomous GPS navigated flight of even unstable vehicles. Datalinks to a ground station, a GPS, remote control and even (brushed) motor controllers are all included in the board which means only 4 motors, a GPS antenna and a frame need to be added to create a nano GPS guided multicopter. Successful GPS

	Specification
Width	20.0 mm
Height	20.0 mm
Weight	2.8 grams
MCU	72MHz 32bit ARM Cortex-M3
RAM	64KB
Flash	512KB
Gyroscope	3 Axis
Accelerometer	3 Axis
Magnetometer	3 Axis
Altimeter	Barometer
GPS	52 channel U-Blox Max-7Q
GPS power Backup	11mF supercapacitor
Power	1S LiPO cell (2.5V-5.5V)
DC motors	4 MOSFET switches
PWM	6 RC servo outputs
UART	1 port
CAN	1 interface
prog./debugging	Serial Wire Debug interface
LED	4 status LED indicators

Table 1: Lisa-S Specifications.

navigated flights were performed using the Walkera Ladybird nano quadrotor frame and replacing its electronics with the Lisa-S.

The open hardware and open software Lisa-S autopilot is hoped to advance the research and development of very light-weight, autonomous MAVs.

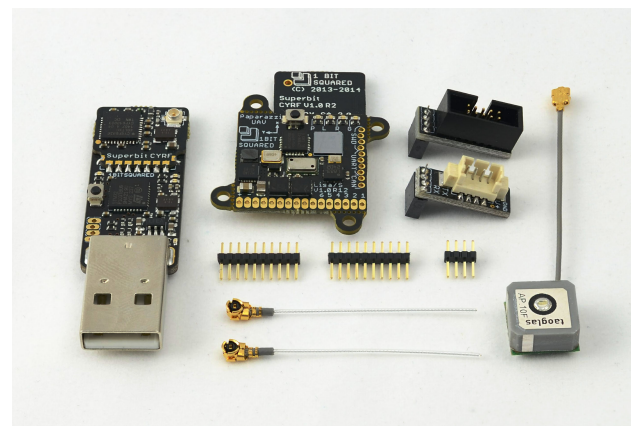


Figure 12: Ladybird starter kit Lisa-S

ACKNOWLEDGEMENTS

We would like to thank all developers of the open source **Paparazzi-UAV** project¹⁴ for their support making this a very powerful project.

¹⁴<http://www.paparazziuav.org/>

Photos in Figures 1-6, 9, 10 and 12 are credit to 1 Bit Squared¹⁵.

REFERENCES

- [1] R.W. Beard, D. Kingston, M. Quigley, D. Snyder, R.S. Christiansen, and W. Johnson. Autonomous vehicle technologies for small fixed-wing uavs. *Journal of Aerospace Computing, Information, and Communication*, 2(1):92–108, 2005. 1
- [2] K.P. Valavanis. *Advances in Unmanned Aerial Vehicles*. Springer, 2007. 1
- [3] H. Chao, Y. Cao, and Y. Chen. Autopilots for small unmanned aerial vehicles: a survey. *International Journal of Control, Automation and Systems*, 8(1):36–44, 2010. 1
- [4] Peter G Ifju, Scott Ettinger, David Jenkins, and Luis Martinez. Composite materials for micro air vehicles. In *INTERNATIONAL SAMPE SYMPOSIUM AND EXHIBITION*, pages 1926–1937. SAMPE, 1999, 2001. 1
- [5] Wei Shyy, Peter Ifju, and Dragos Vieru. Membrane wing-based micro air vehicles. *Applied mechanics reviews*, 58(4):283–301, 2005. 1
- [6] S. Leven, J.-C. Zufferey, and D. Floreano. A minimalist control strategy for small uavs. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2873 –2878, oct. 2009. 1
- [7] C. De Wagter, S. Tijmons, B.D.W. Remes, and G.C.H.E. de Croon. Autonomous flight of a 20-gram flapping wing mav with a 4-gram onboard stereo vision system. In *IEEE Conference on Robotics and Automation (ICRA)*, 2014. 1
- [8] ProxDynamics. Black hornet. <http://www.proxdynamics.com/products/pd-100-black-hornet>. 1
- [9] Bouadi, H. and Simoes Cunha, S. and Drouin, A. and Mora-Camino, F. Adaptive sliding mode control for quadrotor attitude stabilization and altitude tracking. In *Computational Intelligence and Informatics (CINTI), 2011 IEEE 12th International Symposium on*, pages 449 –455, nov. 2011. 1
- [10] Joachim Reuder, Pascal Brisset, Marius Jonassen, Martin Müller, and Stephanie Mayer. The small unmanned meteorological observer sumo: A new tool for atmospheric boundary layer research. *Meteorologische Zeitschrift*, 18(2):141–147, 2009. 1
- [11] Joachim Reuder, P Brisset, M Jonassen, M Müller, and S Mayer. Sumo: A small unmanned meteorological observer for atmospheric boundary layer research. In *IOP Conference Series: Earth and Environmental Science*, volume 1, page 012014. IOP Publishing, 2008. 1
- [12] Pascal Brisset, Antoine Drouin, Michel Gorraz, Pierre-Selim Huard, and Jeremy Tyler. The paparazzi solution. *MAV2006, Sandestin, Florida*, 2006. 1
- [13] Balazs Gati Paparazzi Community. Open source autopilot for academic research the paparazzi system. In *Proceeding of the American Control Conference 2013, Washington, USA*, pages 17–19, June 2013. 1
- [14] MicroPilot. <http://www.micropilot.com/>. 1
- [15] Roland Büchi. *Radio Control with 2.4 GHz*. BoD–Books on Demand, 2014. 3

¹⁵1 Bit Squared <http://1bitsquared.com>

Bibliography

- [1] Mark Jankauski and I.Y. Shen. Dynamic modeling of a rotating insect wing. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 10–17, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-1. [4](#)
- [2] Quoc-Viet Nguyen, Woei Leong Chan, and Marco Debiasi. Design, fabrication, and performance test of an hovering-based flapping-wing micro air vehicle capable of sustained and controlled flight. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 18–25, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-2. [4](#)
- [3] Jaap van de Loosdrecht, Klaas Dijkstra, Jorn Haaije Postma, Wouter Keuning, and Dick Bruin. Twirre: Architecture for autonomous mini-uavs using interchangeable commodity components. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 26–33, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-3. [4](#)
- [4] Woei Leong Chan, Quoc-Viet Nguyen, and Marco Debiasi. Pitch and yaw control of tailless flapping wing mavs by implementing wing root angle deflection. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 34–41, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-4. [4](#)
- [5] Aneesh Chand, Michihiro Kawanishi, and Tatsuo Narikiyo. Design analysis, modelling and experimental validation of a bird-like flapping-wing flying robot. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 42–49, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-5. [4](#)
- [6] Kun Li, Rui Huang, Swee King Phang, Shupeng Lai, Fei Wang, Ping Tan, Ben M. Chen, and Tong Heng Lee. Off-board visual odometry and control of an ultralight quadrotor mav. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 50–57, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-6. [4](#)
- [7] Peng Lu, Erik-Jan van Kampen, and Bin Yu. Actuator fault detection and diagnosis for quadrotors. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 58–63, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-7. [4](#)
- [8] Nina Gaissert, Rainer Mugrauer, Günter Mugrauer, Agalya Jebens, Kristof Jebens, Clemens Rabe, and Elias Maria Knubben. emotionspheres: A new design for human-friendly uavs. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 64–68, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-8. [4](#)
- [9] Joonas Melin, Mikko Lauri, and Risto Ritala. Stereo vision with consumer grade high resolution cameras for a micro air vehicle. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 69–76, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-9. [4](#)
- [10] Marten Schütt, Philipp Hartmann, and Dieter Moormann. Fullscale windtunnel investigation of actuator effectiveness during stationary flight within the entire flight envelope of a tiltwing mav. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 77–83, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-10. [4](#)
- [11] Maarten Verbandt, Bart Theys, and Joris De Schutter. Robust marker-tracking system for vision-based autonomous landing of vtol uavs. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 84–91, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-11. [4](#)

- [12] Qi Wang, Johannes F.L. Goosen, and Fred Van Keulen. Study of design parameters of flapping-wings. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 92–101, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-12. 4
- [13] Sawyer Fuller, Elizabeth Helbling, Pakpong Chirarattananon, and Robert Wood. Using a gyroscope to stabilize the attitude of a fly-sized hovering robot. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 102–109, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-13. 4
- [14] Tijs Noyon, Wee Beng Tay, Bas Van Oudheusden, and Hester Bijl. Effect of chordwise deformation on unsteady aerodynamic mechanisms in hovering flapping flight. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 110–117, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-14. 4
- [15] Matej Karasek, Alexandre Hua, Yanghai Nan, Mohamed Esseghir Lalami, and André Preumont. Pitch and roll control mechanism for a hovering flapping wing mav. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 118–125, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-15. 4
- [16] Yangang Wang, Weixiong Chen, Haitong Wang, and Ruiyu Li. A computational study on the aerodynamics of a 90 mm ducted contra-rotating lift fan. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 126–133, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-16. 4
- [17] Fabrizio Schiano, Javier Alonso-Mora, Konrad Rudin, Paul Beardsley, Roland Siegwart, and Bruno Siciliano. Towards estimation and correction of wind effects on a quadrotor uav. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 134–141, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-17. 4
- [18] Jaysinh Sagar and Arnoud Visser. Obstacle avoidance by combining background subtraction, optical flow and proximity estimation. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 142–149, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-18. 4
- [19] Raheeb Muzaffar and Evsen Yanmaz. Trajectory-aware ad hoc routing protocol for micro aerial vehicle networks. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 150–156, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-19. 4
- [20] Leandro R. Lustosa, François Defay, and Jean-Marc Moschetta. Development of the flight model of a tilt-body mav. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 157–163, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-20. 4
- [21] David Kumar, Akash Sharma, Tigmanshu Goyal, Vemuri Shyam Kumar, P. M. Mohite, and Sudhir Kamle. Deformation analysis of a hummingbird inspired mav flapping wing using digital image correlation. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 164–171, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-21. 4
- [22] Miguel A. Olivares-Mendez, Sumasundar Kannan, and Holger Voos. V-rep & ros testbed for design, test, and tuning of a quadrotor vision based fuzzy control system for autonomous landing. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 172–179, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-22. 4
- [23] Vemuri Shyam Kumar, David Kumar, Tigmanshu Goyal, P M Mohite, and Sudhir Kamle. Development and application of pp-cnt composite for hummingbird inspired mav flapping wings. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 180–187, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-23. 4
- [24] Christian Fuchs, Clark Borst, Guido de Croon, René van Paassen, and Max Mulder. An ecological approach to the supervisory control of uav swarms. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 188–195, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-24. 4

- [25] Ewoud Smeur, Qiping Chu, Guido de Croon, Bart Remes, Christophe De Wagter, and Erik van der Horst. Modelling of a hybrid uav using test flight data. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 196–203, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-25. 4
- [26] Andrea Tenaglia, Mustafa Percin, Bas W. Van Oudheusden, Shuanghou Deng, and Bart Remes. Vortex formation and force generation mechanisms of the delfly ii in hovering flight. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 204–211, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-26. 4
- [27] Sophie Armanini, Jochem Verboom, Guido de Croon, and Coen de Visser. Determination of trim curves for a flapping-wing mav. In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *International Micro Air Vehicle Competition and Conference 2014*, pages 212–217, Delft, The Netherlands, Aug 2014. Paper no. IMAV2014-27. 4
- [28] Muhammad Ridhwan Jumat and Sutthiphong Srigrarom. Design and development of ugs flapping wing mavs. International Micro Air Vehicle Competition and Conference 2014, Delft, The Netherlands, pp. 219–224, Aug 2014. Technical paper. Paper no. IMAV2014-28. 4
- [29] Kirill Shilov. The next generation design of autonomous mav flight control system smartap. International Micro Air Vehicle Competition and Conference 2014, Delft, The Netherlands, pp. 225–229, Aug 2014. Technical paper. Paper no. IMAV2014-29. 4
- [30] Motoki Kimura, Ryosuke Shibasaki, Masahiko Nagai, and Xiaowei Shao. Automatic extraction of moving objects from uav-borne monocular images using multi-view geometric constraints. International Micro Air Vehicle Competition and Conference 2014, Delft, The Netherlands, pp. 230–234, Aug 2014. Technical paper. Paper no. IMAV2014-30. 4
- [31] Sergey Serokhvostov, Nikita Ageev, and Maxim Arkhipov. Design of ducted fans of small height for hexacopter with long hover ability. International Micro Air Vehicle Competition and Conference 2014, Delft, The Netherlands, pp. 235–237, Aug 2014. Technical paper. Paper no. IMAV2014-31. 4
- [32] Alvin Ng, Qiwang Foo, Ridhwan Jumat, Tony Victor Rayar, Wei Chong Koh, and Sutthiphong Srigrarom. Design and build of swarm quadrotor uavs at ugs. International Micro Air Vehicle Competition and Conference 2014, Delft, The Netherlands, pp. 238–246, Aug 2014. Technical paper. Paper no. IMAV2014-32. 4
- [33] Gautier Hattenberger, Murat Bronz, and Michel Gorraz. Using the paparazzi uav system for scientific research. International Micro Air Vehicle Competition and Conference 2014, Delft, The Netherlands, pp. 247–252, Aug 2014. Technical paper. Paper no. IMAV2014-33. 4
- [34] Emmanuel Colles, Fabien Nollet, Bertrand Vandeportaele, and Gautier Hattenberger. Autonomous aerial mapping and observation task for imav2014 competition. International Micro Air Vehicle Competition and Conference 2014, Delft, The Netherlands, pp. 253–258, Aug 2014. Technical paper. Paper no. IMAV2014-34. 4
- [35] Miguel Cordero, Miguel Ángel Trujillo, Jonathan Ruíz, Antonio Jiménez, Luis Díaz, and Antidio Viguria. Flexible framework for the development of versatile mav systems for multi-disciplinary applications. International Micro Air Vehicle Competition and Conference 2014, Delft, The Netherlands, pp. 259–264, Aug 2014. Technical paper. Paper no. IMAV2014-35. 4
- [36] Mario Gäbel, Thomas Krüger, and Ulf Bestmann. Design of a quadrotor system for an autonomous indoor exploration. International Micro Air Vehicle Competition and Conference 2014, Delft, The Netherlands, pp. 265–271, Aug 2014. Technical paper. Paper no. IMAV2014-36. 4
- [37] Samuel Loh, Vishal Vaswani, and Benson Tan. Modular quad rotor (semi-autonomous/autonomous). International Micro Air Vehicle Competition and Conference 2014, Delft, The Netherlands, pp. 272–279, Aug 2014. Technical paper. Paper no. IMAV2014-37. 4
- [38] Bart Remes, Piotr Esden-Tempski, Freek van Tienen, Ewoud Smeur, Christophe De Wagter, and Guido de Croon. Lisa-s 2.8g autopilot for gps-based flight of mavs. International Micro Air Vehicle Competition and Conference 2014, Delft, The Netherlands, pp. 280–285, Aug 2014. Technical paper. Paper no. IMAV2014-38. 4