IMAV2018

10^{th} International Micro Air Vehicle Competition and Conference

November 17-23, 2018

Melbourne, Australia

www.imavs.org RMIT Unmanned Aerial Systems Research Team

Contents

Preface

It is our pleasure to present the proceedings of the 10^{th} International Micro Air Vehicle Conference and Competition, which was held in Melbourne, Australia from November 17 to 23, 2018. This event aims at exchanging 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 48peer-reviewed scientific papers presented at the IMAV 2018. The topics of these papers range from new insights into aerodynamics in buildings to the control of large swarms of MAV with communication limitations. 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.

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.

Nov 2018

IMAV 2018 Program Committee:

S. Watkins
RMIT Unmanned Aerial Systems Research Team

IMAV Organizing Committee

Mark REEDER . . . . Ohio, USA
Bart REMES . . . . Delft, The Netherlands
Gautier HATTENBERGER . . . . Toulouse, France
Jean-Marc MOSCHETTA . . . . Toulouse, France
Simon LACROIX . . . . ?, France
Sergey SEROKHVOSTOV . . . . Moscow, Russia
Henry DE PLINVAL . . . . Toulouse, France
Simon WATKINS . . . . Melbourne, Australia
Dieter MOORMANN . . . . Aachen, Germany
Florian HOLZAPFEL . . . . Munich, Germany
Pascual CAMPOY . . . . Madrid, Spain
Pascal MORIN . . . . ?, France
Sergey SHKARAYEV . . . . Tucson, USA

IMAV2018 Organizing Committee

Technical Committee

Dr Abdulghani Mohamed (Chair)
Dr Matthew Marino
Ms Nicola Kloet
Ms Josephine Phillips
Mr David Tennent
Mr Ashim Panta
Prof Pier Marzocca
Dr Graham Wild

Steering Committee

Prof Simon Watkins (Chair)
Dr Jennifer Palmer
Dr Reece Clothier
Dr Robert Kania
Prof Lakmal Seneviratne

Revision

13261, 2018-11-22 10:38

Sponsors

Supersonic Sponsor

Sonic Sponsors

Subsonic Sponsors

Organisers

Call for Papers

The 10th International Micro Air Vehicle (IMAV2018) Conference and Competition will take place in Melbourne, Australia. The IMAV is a yearly event that combines a scientific conference with a technological competition involving Micro Air Vehicles (MAVs). This is the first time the conference has been held in the Southern Hemisphere and shall be a fantastic opportunity for international research groups to share their knowledge with a global audience. Through the conference we aim to create a forum for academics and practicing engineers to present their latest research findings in the-state-of-art MAV design and application. We encourage authors to present research related to Micro Air Vehicles, research topics may range from fundamental knowledge in disciplinary topics to cross-disciplinary technological innovation and the use of MAVs to other research fields. Topics include, but are not limited to:

- Low Reynolds number aerodynamics
- Navigation in turbulent environments
- Unsteady aerodynamics
- Propulsion set and new energy sources
- Autonomous navigation
- Swarming and formation flight
- Control theory and state estimation
- Flapping wings and bi-inspired MAVs
- Computer vision for MAVs
- Obstacle detection and avoidance
- Integration of MAVs in airspace
- Novel applications for MAVs
- Active perching
- New MAV architectures
- Relayed Control and Communications
- Advanced manufacturing techniques

Exemplary papers may be selected for publishing in the International Journal of MAVS or Unmanned Systems Journal. For your paper to be selected you must register at www.imav2018.org before the start date of the conference. Abstracts should be 1 single-sided page submitted as a word or pdf. file.

Conference Oral Presentations

Day 1 - Morning - Session 1

Day 1 - Morning - Session 2

Day 1 - Afternoon - Session 1

Day 1 - Afternoon - Session 2

Day 2 - Morning - Session 1

Day 2 - Morning - Session 2

Day 2 - Afernoon- Session 1

Day 2 - Afernoon- Session 2

Guest Speakers

Dr. Anya Jones

Anya Jones is an Associate Professor in the Department of Aerospace Engineering at the University of Maryland, College Park, USA. She received her PhD in experimental aerodynamics from the University of Cambridge, United Kingdom, her S.M. in aeronautics and astronautics from MIT, and her B.S. in aeronautical and mechanical engineering from Rensselaer Polytechnic Institute. Her research is focused on the experimental fluid dynamics of unsteady and separated flows. Her current projects focus on wing performance in large-amplitude gust encounters, separated and reverse flow rotor aerodynamics, and flight through airwakes and other unsteady environments. Prof. Jones has been awarded the AFOSR Young Investigator Award (2012), NSF CAREER Award (2016), and the PECASE from the White House (2016). In 2017 she was awarded a Fulbright Scholar Award to the Technion in Haifa, Israel (2017-2018) and an Alexander von Humboldt Research Fellowship to TU Braunschweig in Germany (2018). She is currently chair of a NATO Research Technology Organization task group on gust response and unsteady aerodynamics, an associate fellow of AIAA, and a member of the Alfred Gessow Rotorcraft Center.

Dr. Jones' presentation will focus on MAV behavior in turbulent conditions and the ability to predict unsteady flows and mitigate their effects. One of the challenges of MAV flight is controlled flight through the unsteady environments that exist in urban areas, in airwakes, and in extreme weather. These highly unsteady flows often result in large force transients due to flow separation and the formation of large-scale vortices. The growth and motion of these vortices can have a large impact on the resulting force transient and recovery, necessitating advanced control either locally via flow control or more globally at the vehicle level. The current work focuses on wind gusts and wing maneuvers that result in changes to the relative flow that are of the same order of magnitude as the freestream flow. In these cases, flow separation is significant, so the classical linear solution for the flow does not apply and aggressive control is required. Separated shear layers emanating from the wing tend to roll up into leading and trailing edge vortices that are shed into the wake. The formation and motion of these vortices are characterized via a series of canonical experiments in an attempt to better understand their contribution to aerodynamic forcing and their relative importance as compared to other sources of airloads (e.g., added mass and virtual camber). The results are then used to construct a physics-based low order model of highly separated flows, and thus explore the possibility of predicting unsteady and transient loading, as well as gain insight as to where flow control might be used most effectively to mitigate force transients and/or hasten flow recovery.

Dr. Shane Windsor

Dr Windsor is a Lecturer in Aerospace Engineering at the University of Bristol in the UK. His research is focussed on the fluid dynamics, sensing and control involved in animal flight, and how biological inspiration can be used to improve autonomous systems. Recently his work has focussed on the dynamics of bird flight and how this can inspire the development of technologies for small unmanned air vehicles (UAVs). Previously he worked at Oxford University and the University of Auckland, looking at the biomechanics of a wide variety of different biological systems. He has recently been awarded a prestigious €2 million European Research Council Starting Grant to develop this work further over the next 5 years.

Dr Windsor's talk will focus on his recent work studying the mechanisms behind how birds are such extremely agile and efficient flyers. This will include work looking at:

- the flight stability of gliding birds of prey
- flight path planning in urban wind fields by GPS tagged gulls
- bird manoeuvring flight in windy conditions

The talk will then look at how the insights gained from biological systems can be used to develop technologies for UAVs.

Lieutenant Colonel Keirin Joyce, CSC

Lt. Col. Joyce is currently the UAS Sub-Program Manager for the Australian Army and has been extensively involved with UAS development in the Australian Defence Force for the last 12 years. Lt. Col. Joyce is a graduate of the Australian Defence Force Academy (ADFA) with an Honours Bachelor of Aeronautical Engineering, a Masters in Aviation Management, a Masters of Aerospace Engineering, a Masters in Military and Defence Studies and a Graduate Diploma in Secondary Education (Mathematics), he is currently researching as a Doctorate of Philosophy through ADFA. Lieutenant Colonel Joyce is at the forefront of UAS development in Australia and we look forward to have him attending the conference and competition later this year.

Lt Col. Joyce' will give a talk titled: *The Australian Army's smallest aircraft. Biggest effect?* The Australian Army is the world's largest user of Nano UAS. This talk will discuss why Army has invested in such a tiny aircraft and where they see the future of Micro Air Vehicles.

Dr. Reece Clothier

Dr Clothier is currently a Principal Researcher at Boeing Research & Technology – Australia, the President of the Australian Association for Unmanned Systems (AAUS), and an Honorary Associate Professor at RMIT University. In these roles Dr Clothier has been extensively involved with the UAS development in Australia. Dr Clothier has a Bachelors in Engineering (Aerospace Avionics) and a PhD in the design and certification of UAS from QUT. His research interests include UAS autonomy, certification and safety assurance of highly autonomous systems. In 2016 Dr. Clothier was awarded the Outstanding Next Generation Professional at the Aviation/Aerospace Australia National Awards in recognition of his contribution to the Australian UAS industry. He was also the former Industry Co-chair of the Civil Aviation Safety Authority, Standards Consultative Sub-Committee for Unmanned Systems and currently serves on the General Aviation Advisory Group, which directly advises the Federal Minister of Transport and Infrastructure.

Dr Clothier's presentation is entitled "**The Safety Case Enabling Future Aviation Operational Concepts**"

What will our skies look like in a decade? It is safe to assume that there will be continued growth in the use of Unmanned Aircraft Systems (UAS) for a variety of commercial and civil applications. These applications will be diverse, encompassing point-to-point delivery, high altitude remote data collection and communication services, through to applications in emergency and law enforcement. However, UAS will not be alone. A diversity of new air vehicle concepts providing the community with an entirely new means of personal and mass transportation are expected to emerge. It is becoming apparent that our airspace will not only be much busier but will be used in fundamentally new ways. In the face of such new levels of operational complexity, a key challenge will be in ensuring the risks to other airspace users and the communities overflown can be managed to acceptable levels. This presentation takes a look at the elements of the over-arching safety case enabling the future aviation operational concepts. The different risk factors, and the technical and operational risk mitigations available are presented for a range of UAS and urban air mobility (UAM) concepts of operation. The social and political dimensions influencing community acceptance of the risks are also explored.

Professor. Jason Scholz

Professor Jason Scholz leads research, development and showcasing of high-impact technologies for persistent autonomy, machine cognition, and human-machine integration in close partnership with overseas governments, academia and industry to deliver game changing impact for Australian Defence and National Security. He holds a Professorship (adjunct) in the IT and Engineering Department of UNSW, a PhD in Electrical Engineering from the University of Adelaide and a degree in electronic engineering from the University of South Australia. He has over seventy five refereed publications and patents, in telecommunications, signal processing, artificial intelligence and human decision making. He is an assessor for the Australian Research Council and a graduate of the Australian Institute of Company Directors.

Professor Scholz will speak about the Defence Cooperative Research Centre on Trusted Autonomous Systems. The Minister for Defence industry, the Hon Christopher Pyne MP, has announced that the first Defence Cooperative Research Centre (CRC) will focus on Trusted Autonomous Systems to deliver game-changing unmanned platforms that ensure reliable and effective cooperation between people and machines during dynamic military operations. The CRC for Trusted Autonomous Systems will receive an annual funding of $8 million with a maximum of $50 million over a seven year period. To be effective, Defence needs autonomous systems to be highly trusted, robust and resilient and this initiative will bring together the best researchers from industry and universities to develop the intelligent military platforms of the future. Professor Scholz will discuss the research program and highlight opportunities as it relates to Micro Aerial Vehicles.

Dr Jennifer Palmer

Dr Palmer is a member of the Aerospace Division of the Defence Science & Technology (DST) Group in Melbourne, Australia. Over the past decade, her work has focused on unmanned aircraft, including projects on hybrid-propulsion and power-management technologies for small surveillance aircraft and flapping-wing flight. Currently, Dr Palmer is engaged with academic and DST collaborators in the development of autonomous aircraft for urban missions, including surveillance, humanitarian assistance, and emergency response. Prior to immigrating to Australia from the US, she was employed at Lockheed Martin Missiles & Space in Sunnyvale, California, where her work involved analyses of missile systems and test failures. She earned a Ph.D. in Mechanical Engineering from Stanford University, with a thesis on the demonstration of advanced laser-based diagnostic techniques for hypersonic flows. Dr Palmer's presentation is entitled: **The birds and**

the bees and the MAVs. Nature can provide the inspiration for advances in aircraft technology; and in this talk, several features of biological flyers that may significantly benefit small autonomous aircraft are examined. These include flapping-wing micro air vehicles, fixed-wing aircraft, and rotorcraft, each of which can utilise intelligent control techniques exploited by insects, birds, and bats, such as swarming, gust-response, and energy-harvesting strategies.

Papers

Authors

L

M

N

O

P

Q

R

S

T

V

W

Y

Z

Modeling DelftaCopter from Flight Test Data

J.F. Meulenbeld,* C. De Wagter,† B.D.W. Remes
Delft University of Technology, Kluyverweg 1, 2629HS Delft, the Netherlands

Abstract

The DelftaCopter, a tilt-body tailsitter UAV, endures large gyroscopic moments due to the single helicopter rotor providing its thrust. In previous research by de Wagter et al.[1] the DelftaCopter's attitude dynamics were modeled using a rigid rotor, as is customary for small helicopter modeling. A controller based on this model was unable to compensate coupling between pitch and roll rate caused by gyroscopic moments.

In this paper, two models are compared for reproducing the attitude dynamics of the DelftaCopter in hover. The Cylinder Dynamics (CD) model, used in the previous research, assumes a rigid rotor. The Tip-Path Plane (TPP) model incorporates flapping motion of the blades and was developed by Mettler[2]. The two models are compared by fitting each model's parameters on flight data using chirps, sine waves with increasing frequency, as system identification maneuvers. The TPP model is shown to be much more accurate in reproducing the high-frequency attitude dynamics. An LQR controller directly based on the TPP model is shown to yield adequate tracking performance. This validates the applicability of this model to the DelftaCopter.

For forward flight, an extension to the TPP hover model is proposed incorporating the aerodynamics of the wings and elevons. It is shown that with the extension, chirps in forward flight can be simulated with reasonable accuracy. This paves the way for a model-based controller in this flight state.

1 Introduction

While Unmanned Aerial Vehicles (UAVs) have been around for many years, reaching a large endurance and range on small platforms remains a challenge. The Outback Medical Challenge encourages research on UAVs with the capability of long-distance flights and landing in rough terrain. This requires vertical take-off and landing (VTOL). There are multiple concepts for combining long range and VTOL, of which a qualitative comparison is given by Herbst et al.[3] One of the concepts is the tail-sitter or tilt-body hybrid UAV. This concept has its rotors pointed upwards in hover mode, but can tilt downward by $90°$ to transition to forward flight. In forward flight the wings of the UAV provide the required lift for level flight, which is more efficient than using rotors for lift. The transition is shown in Figure 1.

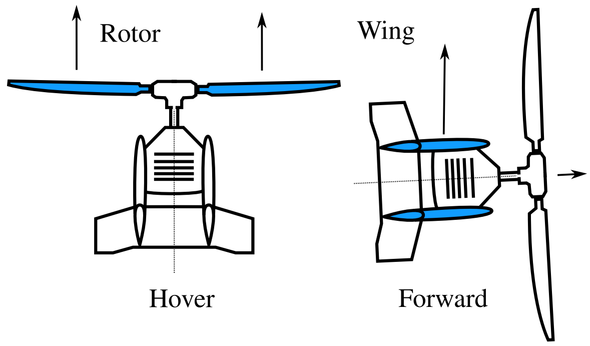


Figure 1: Transition of the DelftaCopter.

Within the class of tilt-body UAVs, the amount of rotors varies. Using four rotors allows to use standard quadcopter control methods in hover, while taking into account aerodynamic forces in forward flight. Examples of this class are the VertiKUL[4] and Quadshot[5]. Two rotor systems have either counter-rotating in-line rotors like the Vertigo[6] or a combination of two rotors that rotate in the same plane, like the MAVIon[7] or Cyclone[8]. Both options use aerodynamic surfaces for control. The single-rotor tilt-body UAV concept is implemented by the DelftaCopter[9] and the Flexrotor, developed by a commercial company[1]. This makes the DelftaCopter a unique platform for researching single-rotor tilt-body UAVs.

The DelftaCopter has been designed and built for the Outback Medical Challenge in 2016 by the DelftaCopter team at TU Delft[9]. In 2018, the new competition again requires a flight of $\approx$ 60 km in one hour, landing in rough terrain halfway[10]. This demands a UAV that has a long range and speed, and can do VTOL. The efficiency advantage of a single rotor over four rotors is why this concept was chosen. The DelftaCopter is shown in Figure 2.

Though the single rotor providing thrust and lift is more efficient than a design with multiple rotors, it yields certain control challenges. As for most helicopters, the gyroscopic effect plays an important role in the dynamics of the DelftaCopter. Contrary to most helicopters, the inertia of the fuse-

*joost.meulenbeld@gmail.com
†C.deWagter@tudelft.nl

[1] `http://aerovel.com/`

Figure 2: DelftaCopter in hover. In forward flight, the entire fuselage pitches down 90° and the rotor provides the thrust as shown in Figure 1.

lage underneath the rotor is quite large compared to the rotor inertia. Often helicopters are controlled with the assumption that an applied roll moment yields a pitch rate and vice versa. This is the case when no bulky fuselage is present under the rotor[11] and the response resembles that of a pure gyroscope. The DelftaCopter has a heavy fuselage due to the long wings and electronics placed at the wing tips for better radio reception. This also makes the inertia much larger in the roll direction than in the pitch direction.

The currently used attitude rate controller was described by De Wagter and Smeur[1]. It uses proportional feedback on the rotational rate error. Additionally, it tries to compensate pitch and roll gyroscopic coupling using identified coupling magnitude $C_{q_{\dot{p}}}$ and $C_{p_{\dot{q}}}$ and another gain K_c. The formulation is given in Equation (1). δ_x is the roll command, δ_y pitch. G is the actuator effectiveness matrix, K_p and K_q are feedback gains, p is roll rate and q is pitch rate, and K_c is a tuning factor.

$$\begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = G^{-1} \begin{bmatrix} K_p p_{err} + q C_{q_{\dot{p}}} K_c \\ K_q q_{err} + p C_{p_{\dot{q}}} K_c \end{bmatrix} \tag{1}$$

This controller relies on the model identified using flight data. The model is based on the assumption of a rigid rotor applying gyroscopic moments. The factor $K_c = 0.5$ in Equation (1) yielded the best results with little coupling between pitch and roll, while a value of 1 would be expected to work best if the underlying model is valid.[1] This indicates that the assumed model is incapable of producing all relevant attitude dynamics of the DelftaCopter. This leads to this paper's research question: What is the best model to replicate the attitude dynamics of the DelftaCopter for the purpose of control?

The research question was answered by comparing multiple models for their accuracy. The results are described in this paper as follows. Section 2 describes the models that are compared. To do this comparison, flight tests were performed and the models' parameter identified, as described in Section 4. With the fitted models, a controller was designed and implemented for hover mode, the design and performance of which is shown in Section 5. In Section 6 an extension to the model is proposed to incorporate the effect of the wing aerodynamics during forward flight. Finally, in Section 7 the results are discussed and a conclusion is drawn with respect to the used models for the DelftaCopter.

2 Identification Modeling

For the purpose of modeling the DelftaCopter for control applications, a system identification model is to be chosen. Not all parameters in this model can be found from direct physical measurement, so a grey-box parameter estimation procedure was used that estimates the unknown parameters from flight data. The models used for this are described in this section.

3 Description of model types

One of the important differences between the assumptions of different models is how they incorporate the flapping dynamics of the helicopter. The flapping dynamics of a helicopter are well covered in Bramwell's helicopter dynamics[12] as follows. The helicopter pitch and roll attitude are controlled by using the swashplate of the rotor. This device allows to set the variation of the blade pitch angle over the rotation of the rotor using collective (for thrust) and cyclic pitch (for attitude control). The blade pitch angle follows the relation below.

$$\theta = \theta_0 - \delta_x \cos\Psi - \delta_y \sin\Psi \tag{2}$$

In this equation taken from Bramwell's book[12, sec. 1.6.2], θ is the blade pitch angle, Ψ is the in-plane rotation of the blade from the back of the helicopter and θ_0 is the collective pitch angle determining the lift generated by the rotor. δ_x and δ_y are determined by the cyclic pitch setting of the (auto)pilot. This variation of pitch angle over the rotation will generate differences in aerodynamic forces which in turn makes the blade flap up and down. The following categories of models can be distinguished based on how they incorporate the flapping dynamics.

- The first and most elaborate approach is to deal with the flapping blade explicitly, including the flapping angle of every individual blade as a state of the model. This allows theoretical analysis into the response but also requires many physical parameters to be accurately measured. It leads to a model that is very hard to use for control design due to its time-dependence.

- An often applied simplification is to relate attitude dynamics to the Tip-Path Plane (TPP), i.e. the plane in which the tips of the rotor travel. It can be derived directly using the flapping angle equation shown in Equation (2). A mathematical description is derived by Mettler by neglecting high-frequency dynamics of the rotor.[2] The TPP is represented by two angles, a and b, which indicate the angle between the TPP and the horizontal, in the longitudinal and lateral direction respectively, as shown in Figure 3. The TPP angles change under influence of control inputs, rotations of the fuselage and gyroscopic precession of the rotor. A moment is applied on the fuselage if the a and b angles are not zero, due to the effective spring between the rotor blade and axis and the offset of thrust application point on the rotor and the center of mass of the fuselage.[2, 13] This way, the rotor is a separate body from the fuselage with its own dynamics. This simplification is valid if the rotor rotational frequency ($\approx$ 1650RPM $=$ 27.5 Hz for the DelftaCopter) is much higher than the highest eigenfrequency of the body-TPP coupling dynamics ($\approx$ 5 Hz for the DelftaCopter, as shown later). Then the oscillations of the rotor are damped out by the body dynamics and the forces can be expressed as an average over a rotation of the rotor. Mettler applies this type of model to a small unmanned helicopter, resulting in an accurate model that matches with flight data[2].

- The last simplification is to ignore the flapping dynamics and treat the rotor as a rigid object, with no flapping angle possible. This way, the gyroscopic effect that the rotor produces still affects the fuselage, but the time-dependent coupling between fuselage and rotor are eliminated. This method is used widely for small helicopters[14, 15, 16]. A TPP model can be transformed into a model without flapping dynamics by setting the derivatives of the a and b angles to zero, which is a valid simplification if the flapping dynamics are much faster than the fuselage dynamics[13]. The type of model without flapping dynamics will be called a cylinder dynamics (CD) model, since the gyroscopic effect of the rotor can be included by modeling it as a rigid rotating cylinder.

In this paper, the TPP and CD models are compared for their accuracy to replicate in-flight test data. The formulation for the TPP and CD model will be given below. Both are linear time-invariant models in state-space form, using the equations below.

$$\dot{\bar{x}} = A\bar{x} + B\bar{u} \tag{3}$$

$$\bar{y} = C\bar{x} + D\bar{u} \tag{4}$$

3.1 TPP model description

The Tip-Path plane (TPP) model has been adapted from Mettler's helicopter model, of which only the p, q, a and b states are used[2]. For the TPP model, the state vector is $\bar{x} = (p, q, a, b)^T$ and the input vector is $\bar{u} = (\delta_x, \delta_y)^T$. The state-space A, B and C matrices for the TPP model are given in Equations (5) to (7). The D-matrix consists of only zeros. The model has nine parameters that need to be found from flight testing: L_b and M_a represent the spring constants of the Tip-Path Plane, consisting of both the stiffness of the blade and blade hinge, and the offset between rotor and center of mass of the fuselage. τ_{f_n} is the time constant of the TPP dynamics. A_{b_n} and B_{a_n} are cross-coupling terms that describe how the TPP interchanges the a and b angles over time. The four parameters in the B-matrix give the actuator effectiveness. Since the DelftaCopter may fly with different RPMs, the theoretical dependence of the parameters is made explicit: $\tau_f = \tau_{f_n}/\Omega$, $A_b = A_{b_n}/\Omega^2$ and $B_a = B_{a_n}/\Omega^2$[2, sec. 2.3]. Section 4.3 comments on the validity of this dependence.

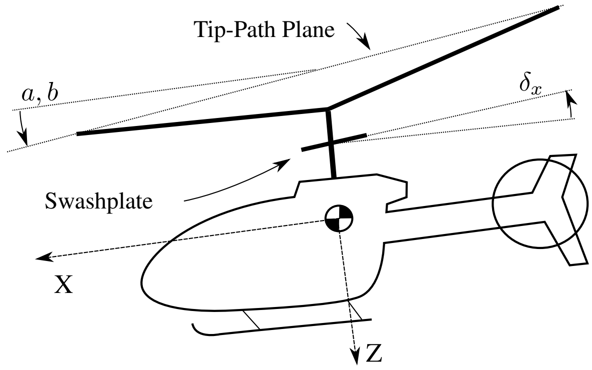


Figure 3: Helicopter Tip-Path plane (TPP) model. The axes shown in this image are as used in the DelftaCopter while it is in hover. The y-axis is chosen using the right-hand rule.

$$A_{TPP} = \begin{bmatrix} 0 & 0 & 0 & \mathrm{L_b} \\ 0 & 0 & \mathrm{M_a} & 0 \\ 0 & -1 & -\frac{\Omega}{\tau_{f_n}} & \frac{\mathrm{A_{b_n}}}{\Omega\,\tau_{f_n}} \\ -1 & 0 & \frac{\mathrm{B_{a_n}}}{\Omega\,\tau_{f_n}} & -\frac{\Omega}{\tau_{f_n}} \end{bmatrix} \tag{5}$$

$$B_{TPP} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{\mathrm{A_{lat}}\,\Omega}{\tau_{f_n}} & \frac{\mathrm{A_{lon}}\,\Omega}{\tau_{f_n}} \\ \frac{\mathrm{B_{lat}}\,\Omega}{\tau_{f_n}} & \frac{\mathrm{B_{lon}}\,\Omega}{\tau_{f_n}} \end{bmatrix} \tag{6}$$

$$C_{TPP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{7}$$

3.2 CD model description

The state used in the Cylinder Dynamics (CD) model is $\bar{x} = (p, q)^T$, and the input is the same as for the TPP model. Both the A- and B-matrices consist of four identifiable parameters, as shown in Equations (8) and (9) below. All identifiable variables could be ascribed physical meaning as a steady-state solution of the a and b states in the TPP model, or through gyroscopic moments and aerodynamic damping. The latter option would entail measuring many different parameters including aerodynamic forces, and does not improve usefulness for control design since the final model has the same structure as it has now.

$$A_{cyl} = \begin{bmatrix} L_p & L_q \\ M_p & M_q \end{bmatrix} \tag{8}$$

$$B_{cyl} = \begin{bmatrix} L_{lat} & L_{lon} \\ M_{lat} & M_{lon} \end{bmatrix} \tag{9}$$

3.3 *TPP and CD model equivalency*

The a and b states of the TPP model can be regarded as the angular accelerations in q and p states respectively, i.e. the a state is directly related to the derivative of q. This means that instead of determining the angular acceleration directly using the control input as in the CD model, the angular acceleration has its own dynamics and is influenced by the control input.

The steady-state solution resulting from $\dot{a} = \dot{b} = 0$ in the TPP models allows substitution of steady-state a and b values in the $\dot{p}$ and $\dot{q}$ equations, which then yields a comparable system to the CD model, with every CD model parameter linked directly to a combination of parameters in the TPP model.[13]

3.4 *Conclusion*

The CD and TPP models will be fitted purely based on the input-output response. For the TPP model the a and b states are not measured, but result from the measurements of their interactions with the pitch and roll rates and actuators. This means that these states cannot directly be validated from measurements. The TPP model has but one parameter more than the CD model, but more importantly it has two more eigenfrequencies that should allow capturing a broader range of response frequencies.

4 Flight testing and system identification

For fitting the model parameters, flight tests were performed in an indoor environment. To make sure that the range of interesting dynamics is covered in every flight test, an automated flight testing procedure was developed. According to Tischler et al., the "chirp" maneuver is able to generate the required frequency content.[17] A chirp is a sine wave with a frequency increasing continuously over time. When a chirp is inserted in a single actuator, the system response shows the coupling between all axes and how this changes with frequency. Tischler also states that to get the best results, the system should be flown open-loop. However, this is not possible with the DelftaCopter due to the dynamics, so the attitude controller is still active during flight tests. To lower the coherence between different axes introduced by the controller, noise is added to every axis independently, as per Tischler's suggestion[17]. The noise is filtered with a first-order low-pass filter with the cut-off at the highest frequency of the chirp. An exponential-time chirp is used to have enough content at the lower frequencies.

The chirp signal is generated and added to the controller signal and the resulting actuator signal is stored with the gyroscope measurements on an SD-card at a frequency of 500 Hz. The chirp settings are given in Table 1. The frequency range starts and ends higher than suggested by Mettler[2]. The lower frequency is limited by the size of the indoor facility in which the tests were performed, while the higher frequency now includes more of the high-frequency dynamics. The eigenfrequencies of the identified model are well within the range of the chirp.

Variable	Value
Start frequency	0.5 Hz
End frequency	10 Hz
Noise fraction	0.2
C_1	4
C_2	$\frac{1}{\exp(C_1)-1}$

Table 1: Settings used for the exponential-time chirp. The noise fraction is the ratio between the amplitude of the chirp and the standard deviation of the white noise that is filtered and added to the chirp signal. C_1 and C_2 are the values used in the exponential-time chirp formulation in the book by Tischler[17].

After flight testing, the data was filtered digitally by an ideal low-pass filter with a cut-off frequency of 15 Hz. This removes vibrations caused by the rotor which rotates at around 27.5 Hz. The input channels were centered around 0 to remove input bias. The resulting data streams were used to fit the parameters in the time domain using the MATLAB system identification toolbox.

4.1 *Chirp results*

The results for a roll chirp are shown in Figure 6. In this figure, the measured roll and pitch rates are shown for the online measurement, the simulated TPP model and simulated CD model. The pitch rate q is a result of the pitch-roll coupling introduced by the rotor. It is clear that the CD model is able to accurately simulate the response up to a certain frequency, but does not include the eigenfrequency which is excited at around 28.5 s. The TPP model does include this eigenfrequency and is much better at reconstructing the system response. This is due to the fact that the TPP dynamics model has four states, and therefore four eigenfrequencies. The two eigenfrequencies the TPP model has extra compared to the CD model lead to higher frequencies being accurately modeled as well. A pitch chirp is shown in Figure 7. In this chirp, the mismatch between the CD model and measurements is even worse.

4.2 *Model fit validation*

Figure 4 shows pitch and roll doublets, as flown by a pilot while the attitude controller was active. This flight data was not used for fitting the models and can thus be used as validation for model accuracy. Table 2 shows the eigenfrequencies of both the identified TPP and CD systems. The slower eigenfrequency corresponds to a pitch-roll coupled motion and is present in both models. The faster eigenfrequency is almost purely pitch and is not present in the CD model. This explains why the pitch response of the CD model is so far off.

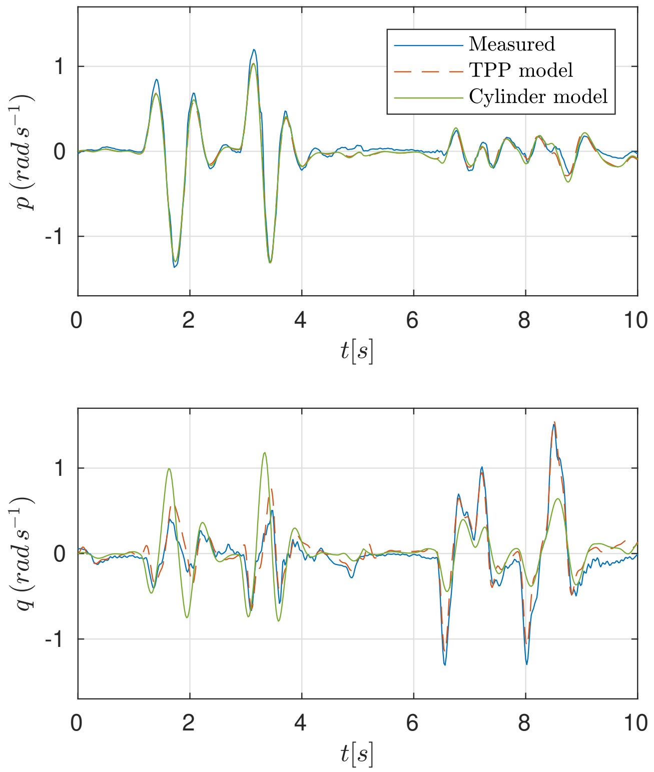


Figure 4: First some roll, then some pitch doublets, flown manually with the attitude controller active. While the roll response is quite accurately modeled by both models, the pitch response is much better in the TPP model. Since the CD model misses the higher eigenfrequency, the fast movements of the pitch are not accurately modeled. This difference is due to the low pitch inertia compared to the roll inertia.

The numerical difference between the measurements and the model output is given in Table 3. The Normalized Root Mean Squared Error (NRMSE) percentage is used as a measure of the goodness of fit, where 100% constitutes a perfect match. The NRMSE percentage is given in Equation (10), where y is the measured signal, $\hat{y}$ is the model output and $\bar{y}$ is the average of the measured signal. The fraction in Equation (10) is thus also equal to the root mean squared error divided by the standard deviation of the measurement.

$$NRMSE = 100\% \left(1 - \frac{||y - \hat{y}||}{||y - \bar{y}||}\right) \tag{10}$$

The fitted parameters are shown in Table 4. The difference in roll and pitch inertia is apparent from the L_b and M_a values, which differ by a factor of 4.8. To validate that the parameters were not overfitted to a particular chirp, the TPP model was fitted to two different sets of chirp data and their fitted parameters were compared. The highest single change in model parameter between the two sets of chirps was 7.7%, but the eigenfrequencies and damping ratios of the systems differ at most 0.9% and 1.8% respectively.

	Pole	Frequency [Hz]	Damping [−]
CD model	1-2	1.54	0.35
TPP model	1-2	1.64	0.39
	3-4	5.04	0.22

Table 2: Comparison of eigenfrequencies of the CD model and flapping dynamics fitted models. The first pole-pair has almost the same eigenfrequency between the two models, which means that in the lower frequencies both models respond comparable. However, the higher eigenfrequency of the flapping dynamics model is not present in the CD model, which explains why high-frequency dynamics are completely damped out in these simulations, as shown in Figure 7.

	Axis	TPP model	CD model
Chirp	p	77.8	77.2
	q	77.3	25.9
Doublets	p	77.6	76.7
	q	64.7	20.0

Table 3: Comparison of NRMSE percentage as given in Equation (10). For both the chirp and doublet value the roll response of both models is similar, but the pitch response matches measurements much better with the TPP model. Still, the pitch response is not perfect, as can also be seen in the doublet time response in Figure 4. The chirp rows show the NRMSE percentage for two pitch and two roll chirps combined while the doublets rows for two pitch and two roll doublets.

4.3 Parameter RPM dependence

As stated in Section 2, the dependence of the parameters on RPM has been made explicit in the system identification model. Therefore the remaining identifiable parameters should remain constant for different RPMs. A range of RPMs between 1500 and 1650 was tested, and the parameters, which should stay constant, change up to 185% with many parameters changing tens of percentage points. To analyse how the actual model characteristics have changed, the RPM parameter Ω is changed to 1650 RPM in the A- and B-matrices of the model fitted on the 1500 RPM data. If the theoretical relations are correct, the resulting eigenfrequencies and damping should be equal to the model fitted directly on the 1650 RPM chirp data. In reality, the largest eigenfrequency change was 4%, while the largest damping ratio change is 28%, making the model response substantially different from expected. The model fitted on 1500 RPM yielded NRMSE percentages of 72.4% and 67.1% for roll and pitch axes respectively, when tested on the same chirp signal as used in Table 3. Multiple controllers based on models at different RPMs would proba-

Param	Value	CD	Value
A_b	-1.338	L_p	-2.056
B_a	1.448	M_p	10.536
L_b	147.548	L_q	-7.900
M_a	713.378	M_q	-4.777
τ_f	0.091	L_{lat}	-5.361
A_{lat}	-0.282	M_{lat}	-67.573
A_{lon}	0.296	L_{lon}	9.917
B_{lat}	0.524	M_{lon}	11.136
B_{lon}	-0.050		

Table 4: The fitted values for the TPP and CD models. The variables are those given in Equations (5), (6), (8) and (9), while for the TPP model the variables have been made dependent on the RPM again by using the substitutions $\tau_f = \tau_{f_n}/\Omega$, $A_b = A_{b_n}/\Omega^2$ and $B_a = B_{a_n}/\Omega^2$.

bly be needed to accurately control the DelftaCopter. Further research is thus needed to obtain models that generalize well with different rotor RPMs.

4.4 Conclusion

It is clear that the TPP model is much more accurate than the CD model for the DelftaCopter. The extra complex eigenfrequency pair allows better dynamics resolution at higher frequency. Validation doublets confirm this result.

5 Rate control in hover

In order to further test the validity of the model, a rate controller was designed using standard control techniques and tested. Since the a and b angles are not measured, a linear observer was used to estimate these states in real time, of the form as given in Equation (11). $\hat{x}$ is the current state estimate and L is the correction matrix. The A, B and C matrices are as given in Equations (5) to (7).

$$\dot{\hat{x}} = A\hat{x} + B\bar{u} + L(\bar{y}_{measured} - \hat{y}) \quad (11)$$

$$\hat{y} = C\hat{x} + D\bar{u} \quad (12)$$

The L matrix is chosen using pole placement, setting the poles of the observer at $(-50, -50, -51, 51)$. This is small enough to add some damping to the vibrations caused by the rotor on the gyroscope readings.

The controller is designed using the feedback law given in Equation (13), adding the reference attitude rate $\bar{y}_{ref}$ multiplied by the steady-state gain of the controlled system g, which is given in Equation (14).

$$\bar{u} = -K\hat{x} + g\bar{y}_{ref} \quad (13)$$

$$g = (C(-A + BK)^{-1})^{-1} \quad (14)$$

The gain matrix K is chosen using LQR. This technique finds the optimal gain matrix K for the system minimizing a cost function of state and inputs. The cost matrices of the LQR design were chosen such that the a and b state cost is very low at 0.001, since these states are not important to the end goal of stabilizing and controlling the attitude rate. The p and q states were given equal cost, fixed at 1. Controllers were then designed for different costs of the system inputs, yielding controllers that are more or less aggressive depending on the input cost. The lower the cost on the input, the faster the controller steers the system, up to the point where input lag and delay makes the response oscillatory and unstable. The input cost of 5 made the system the fastest without introducing these oscillations. The controlled system response is shown in Figure 5. It can be seen that the roll response is delayed, but the measurement follows the command quite well. The pitch response shows a coupling when larger roll rates are present. The pitch response signal is also larger in magnitude than the commanded rates.

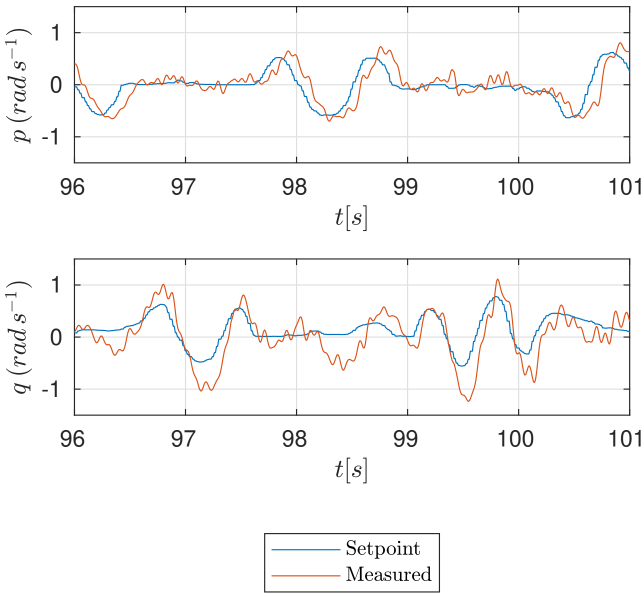


Figure 5: Controller performance during a piloted flight. The pilot directly commands the attitude rate. The roll response is delayed with respect to the command but otherwise has adequate tracking performance. The pitch rate shows coupling with the roll and the response shows larger values than the commands.

The fact that standard control techniques can be used to design a controller for the DelftaCopter, confirms that the TPP model is applicable. While the response is not perfect, the controller is able to stabilize the DelftaCopter without other tuning parameters.

6 Forward Flight Modeling

In forward flight, the DelftaCopter pitches down 90° such that the wings are level with the ground. The airspeed increases and the wings generate the required lift to maintain altitude, while the main rotor head is now providing thrust for the aircraft. This means that the aerodynamic surfaces play a significant role in the balance of forces and moments of the DelftaCopter. To be able to design a controller for this flight mode, the TPP and CD hover models were extended. During the forward flight mode, the roll angle of the DelftaCopter in hover mode constitutes a yaw angle in the traditional aircraft sense, but is still referred to as roll. Roll rate (yaw rate in standard aircraft reference frames) is still denoted p. In order to come to a linear model, linear aerodynamic moments are assumed. The model is fitted on forward flight data and a conclusion is drawn on how significant the parameters are.

First of all, the DelftaCopter has four movable surfaces that together supply the role of aileron and elevator where every surface deflection is a linear combination of the aileron and elevator commands. The aileron applies moments to the fuselage along the axis of rotation of the rotor, and as such does not induce coupling on the pitch and roll axes. The elevator does induce a moment on the pitch rate and is thus included in the model with parameter M_{elev}. The short-period longitudinal damping due to a pitch rate M_q and lateral side-slip damping due to a roll rate L_p are included. In the CD model, they are lumped into the parameters already present, while in the TPP model these damping parameters are not yet present and are added. The resulting $A_{TPP,FW}$ and $B_{TPP,FW}$ matrices for forward flight are given in Equations (15) and (16). The state vector is the same as for the hover vector while the input vector now is $\bar{u} = (\delta_x, \delta_y, \delta_e)^T$.

$$A_{TPP,FW} = \begin{bmatrix} \mathrm{L_p} & 0 & 0 & \mathrm{L_b} \\ 0 & \mathrm{M_q} & \mathrm{M_a} & 0 \\ 0 & -1 & -\frac{\Omega}{\tau_{f_n}} & \frac{\mathrm{A_{b_n}}}{\Omega\,\tau_{f_n}} \\ -1 & 0 & \frac{\mathrm{B_{a_n}}}{\Omega\,\tau_{f_n}} & -\frac{\Omega}{\tau_{f_n}} \end{bmatrix} \tag{15}$$

$$B_{TPP,FW} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & M_{elev} \\ \frac{\mathrm{A_{lat}}\,\Omega}{\tau_{f_n}} & \frac{\mathrm{A_{lon}}\,\Omega}{\tau_{f_n}} & 0 \\ \frac{\mathrm{B_{lat}}\,\Omega}{\tau_{f_n}} & \frac{\mathrm{B_{lon}}\,\Omega}{\tau_{f_n}} & 0 \end{bmatrix} \tag{16}$$

L_p and M_q represent aerodynamic damping on the roll and pitch rate. As before, the parameters L_b and M_a of the TPP model are the representative spring constants, and their physical meaning is given by Mettler[2, sec. 3.1], with M_a for example given in Equation (17). This relates the TPP angle to the angular acceleration of the fuselage, and contains a rotor stiffness term and a thrust term. The rotor blade spring stiffness k_β is equal to $88\,\mathrm{N\,m\,rad^{-1}}$[1], which is much higher than the thrust contribution at maximum weight $hT \approx hmg \approx 0.15 \cdot 4.5 \cdot 9.81 = 6.6\,\mathrm{N\,m\,rad^{-1}}$. Therefore, while the thrust may become smaller in forward flight, the parameters L_b and M_a are assumed constant in fitting the forward flight models.

$$M_a = \frac{k_\beta + hT}{I_{yy}} \tag{17}$$

The flight test used to fit the parameters is a forward flight test in level flight. The airspeed fluctuates between $17\,\mathrm{m\,s^{-1}}$ and $19.5\,\mathrm{m\,s^{-1}}$, while the RPM fluctuates between 1550 and 1720. This RPM range is quite broad compared to the hover experiments, especially considering the observed sensitivity of parameters to the RPM.

The parameters of the TPP and CD models are given in Table 5. Comparing these to the hover parameters as given in Table 4 shows that the hover and forward flight models have comparable parameters for the rotor dynamics in the TPP model case. It seems that the roll rate damping is very small, which is logical since the vertical stabilizer has a small moment arm to the center of mass.

TPP	Value	CD	Value
$\mathrm{A_{b_n}}$	−0.908	L_p	−10.690
$\mathrm{B_{a_n}}$	0.999	M_p	14.899
$\mathrm{L_b}$	147.550	L_q	−9.251
$\mathrm{M_a}$	713.380	M_q	1.050
τ_{f_n}	0.075	L_{lat}	6.605
$\mathrm{A_{lat}}$	−0.196	M_{lat}	−70.459
$\mathrm{A_{lon}}$	0.214	L_{lon}	−2.903
$\mathrm{B_{lat}}$	0.440	M_{lon}	11.532
$\mathrm{B_{lon}}$	−0.026	M_{elev}	10.263
$\mathrm{L_p}$	−0.930		
$\mathrm{M_q}$	4.691		
$\mathrm{M_{elev}}$	37.752		

Table 5: The fitted values for the TPP and CD models in forward flight. The variables for the TPP model have been made dependent on the RPM again by using the substitutions $\tau_f = \tau_{f_n}/\Omega$, $A_b = A_{b_n}/\Omega^2$ and $B_a = B_{a_n}/\Omega^2$. The $A_{TPP,FW}$ and $B_{TPP,FW}$ matrices of the forward TPP model are given in Equations (15) and (16). The CD model $A_{cyl,FW}$-matrix is the same as Equation (8), while the $B_{cyl,FW}$ matrix is as given in Equation (8), with the addition of a third input, elevator δ_{elev} linearly related to pitch acceleration $\dot{q}$ through parameter M_{elev}.

In Figures 8 to 10 the models are simulated on measurements of forward flight data, on roll, pitch and elevator chirps respectively. This chirp data was not used for fitting the parameters and can be used for validation. It is clear that the TPP model is better at predicting the high-frequency response than the CD model, but the fit is not as good as on the hover mode. This can be due to the RPM fluctuations or the TPP model not being applicable to the high rotor inflow experienced in forward flight. Another cause of model inaccuracy could be aerodynamic effects missing in the model. The angle of attack is not part of the model and was not measured, but could have an important influence. Surprisingly, the M_q parameter is positive, implying a positive feedback loop on the pitch rate. This could be due to the missing other influences in the model. The accuracies of the models can be seen in Table 6.

	Axis	TPP model	CD model
Fitting	p	66.0	70.8
	q	54.7	21.7
Validation	p	49.7	53.1
	q	47.1	17.0

Table 6: Comparison of NRMSE percentage as given in Equation (10) for forward flight. Both fitting and validation percentages concern three chirps, one roll, one pitch and one elevator. Simulation of the models on the validation chirps can be found in Figures 8 to 10.

7 Conclusion

In order to design a new controller for the DelftaCopter, a new modeling approach was required that better captures the attitude dynamics. A system identification modeling approach was chosen and two models were compared: the previously used Cylinder Dynamics (CD) model which assumes a rigid rotor[1], and a Tip-Path Plane (TPP) model which was derived by Mettler[2]. Chirps were used as system identification maneuver and the models were compared. It is clear from Section 4 that the TPP model is much better at generating the high-frequency response than the CD model. This is validated using manually flown doublets, for which the TPP response also shows better accuracy. To show that this model is usable for control design, an attitude rate controller is designed using the standard LQR technique, with reasonable control response as shown. It can thus be concluded that the flapping dynamics have a significant influence on the attitude dynamics of the DelftaCopter.

The relationship between the identified parameters and rotor RPM was found to be different than predicted from theory. This may be due to the lumping together of unmodeled effects into the present parameters. Probably models at different RPMs are required for accurate control at these different RPMs.

For forward flight, the TPP model was extended to include roll rate and pitch rate damping, while both the TPP model and CD model include a constant for the elevator effectiveness. It is shown that the hover model with this extension is applicable to forward to some extent.

8 Recommendations

The following recommendations are made on what research could be conducted next:

- Measure the a and b states in flight to give more accurate models and validate the current model's prediction.
- Investigate the TPP model's parameters' dependence on RPM to broaden the accurate flight envelope of the model.
- Measure more aircraft states in forward flight, to allow the attitude dynamics model to depend on, in particular, angle of attack. Generalize the forward flight model for different airspeeds and RPMs.
- Use the forward flight model for attitude control.

Acknowledgements

The author would like to thank Kevin van Hecke, Freek van Tienen and Erik van der Horst from the DelftaCopter team for all help with doing flight tests and Bart Slinger and Willem Melching for the discussions on control theory.

References

[1] Christophe De Wagter and Ewoud J.J. Smeur. Control of a hybrid helicopter with wings. *International Journal of Micro Air Vehicles*, 9(3):209–217, sep 2017.

[2] Bernard Mettler. *Identification Modeling and Characteristics of Miniature Rotorcraft*. Springer US, Boston, MA, 2003.

[3] S Herbst, G Wortmann, and M Hornung. Conceptual design studies of vertical takeoff and landing remotely piloted aircraft systems for hybrid missions. *CEAS Aeronautical Journal*, 7:135–148, 2016.

[4] Menno Hochstenbach, Cyriel Notteboom, Bart Theys, and Joris De Schutter. Design and Control of an Unmanned Aerial Vehicle for Autonomous Parcel Delivery with Transition from Vertical Take-off to Forward Flight -VertiKUL, a Quadcopter Tailsitter. *International Journal of Micro Air Vehicles*, 7(4):395–405, 2015.

[5] Atsushi Oosedo, Satoko Abiko, Atsushi Konno, Takuya Koizumi, Tatuya Furui, and Masaru Uchiyama. Development of a quad rotor tail-sitter VTOL UAV without control surfaces and experimental verification. In *2013 IEEE International Conference on Robotics and Automation*, pages 317–322. IEEE, may 2013.

[6] B. Bataillé, J.-M. Moschetta, D. Poinsot, C. Bérard, and A. Piquereau. Development of a VTOL mini UAV for multi-tasking missions. *The Aeronautical Journal*, 113(1140):87–98, feb 2009.

[7] L.R. Lustosa, F. Defay, and J.M. Moschetta. Development of the flight model of a tilt-body MAV. In *IMAV*, jan 2014.

[8] M Bronz, EJ Smeur, and HG de Marina. Development of A Fixed-Wing mini UAV with Transitioning Flight Capability. *35th AIAA Applied*, 2017.

[9] Christophe De Wagter, Rick Ruijsink, Ewoud Smeur, Kevin van Hecke, Freek van Tienen, Erik van der Horst, and Bart Remes. Design, Control and Visual Navigation of the DelftaCopter. jan 2017.

[10] UAV Challenge. UAVC Medical Express 2018 Rules V2, 2017.

[11] Konstantin Kondak, Markus Bernard, Nikolas Losse, and Günter Hommel. Autonomously Flying VTOL-Robots: Modeling and Control. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 736–741. IEEE, apr 2007.

[12] A.R.S. Bramwell, G. Done, and David. Balmford. *Bramwell's Helicopter Dynamics*. American Institute of Aeronautics and Astronautics, 2001.

[13] Bernard Mettler, Chris Dever, and Eric Feron. Scaling Effects and Dynamic Characteristics of Miniature Rotorcraft. *Journal of Guidance, Control, and Dynamics*, 27(3):466–478, may 2004.

[14] Konstantin Kondak, Markus Bernard, Nikolas Losse, and Günter Hommel. Elaborated modeling and control for autonomous small size helicopters. In *Proc. ISR/ Robotik*, 2006.

[15] Subodh Bhandari, Richard Colgren, Philipp Lederbogen, and Scott Kowalchuk. Six-DoF dynamic modeling and flight testing of a UAV helicopter. *AIAA Modeling and Simulation Technologies Conference and Exhibit*, (August):992–1008, 2005.

[16] Luis A Sandino, Manuel Bejar, Anibal Ollero, L A Sandino, A Ollero Grvc, M Bejar, and A Ollero. A Survey on Methods for Elaborated Modeling of the Mechanics of a Small-Size Helicopter. Analysis and Comparison. *Journal of Intelligent & Robotic Systems*, 72:219–238, 2013.

[17] Mark B Tischler and Robert K. Remple. *Aircraft and Rotorcraft System Identification: Engineering Methods with Flight Test Examples*. 2008.

APPENDIX A: CHIRP MEASUREMENTS

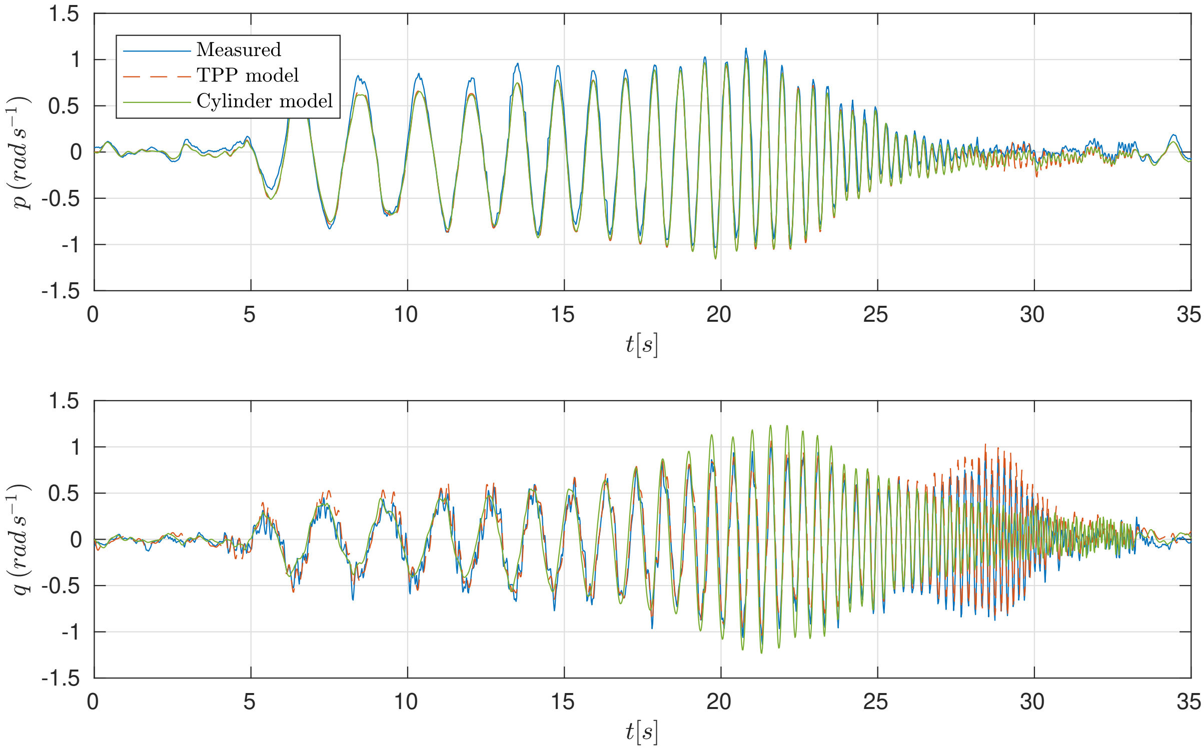


Figure 6: Chirp on the roll axis in hover. The pitch motions are mainly due to pitch-roll coupling. The TPP model is much better at producing the measured pitch signals in the higher frequency range.

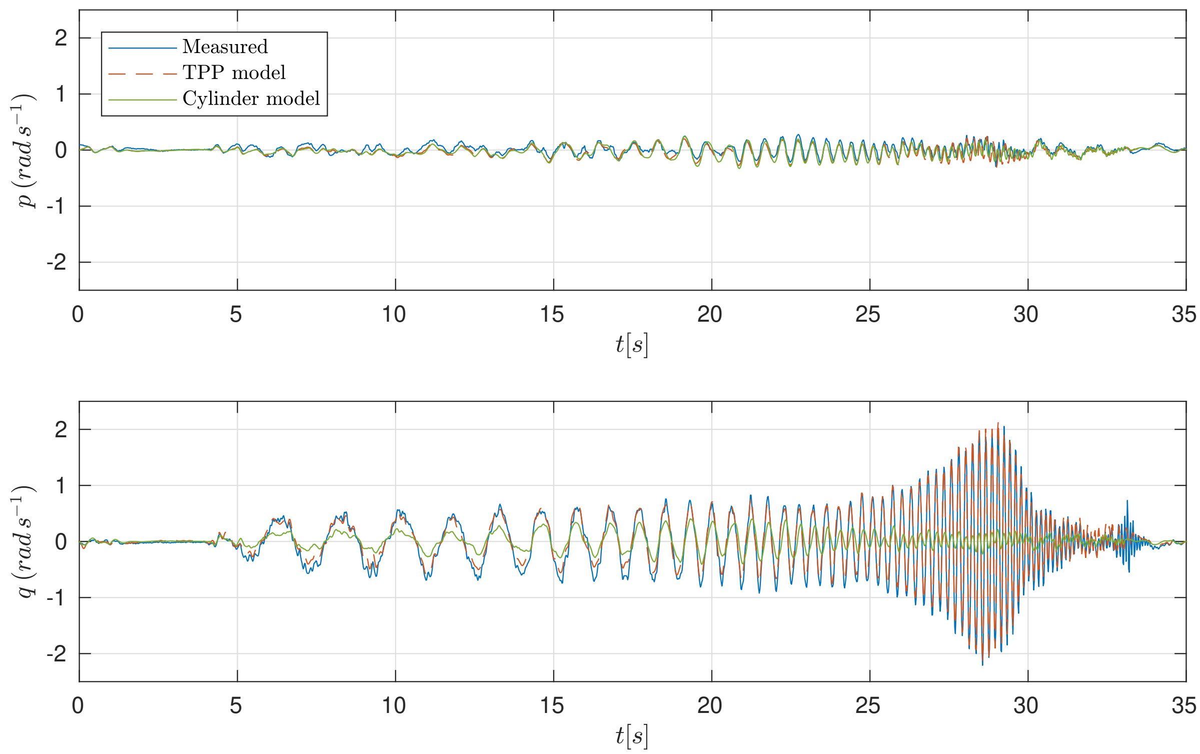


Figure 7: Chirp on the pitch axis in hover. The roll motion due to pitching is much less severe due to the high roll inertia compared to the pitch inertia. Now the CD model's accuracy at low frequencies is also worse than the TPP model.

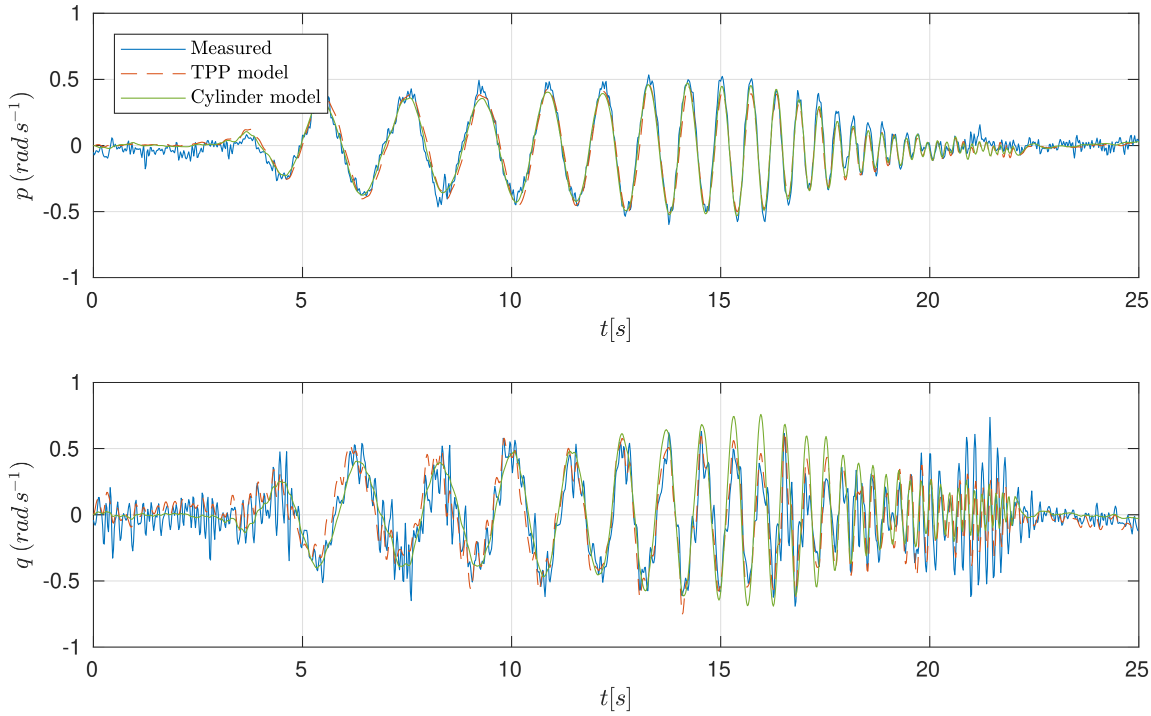


Figure 8: Validation chirp on the roll axis in forward flight. The pitch motions are probably mainly due to pitch-roll coupling. The TPP and CD model have a similar response. The high-frequency fluctuations in the pitch response are due to the attitude controller which is active during the chirp.

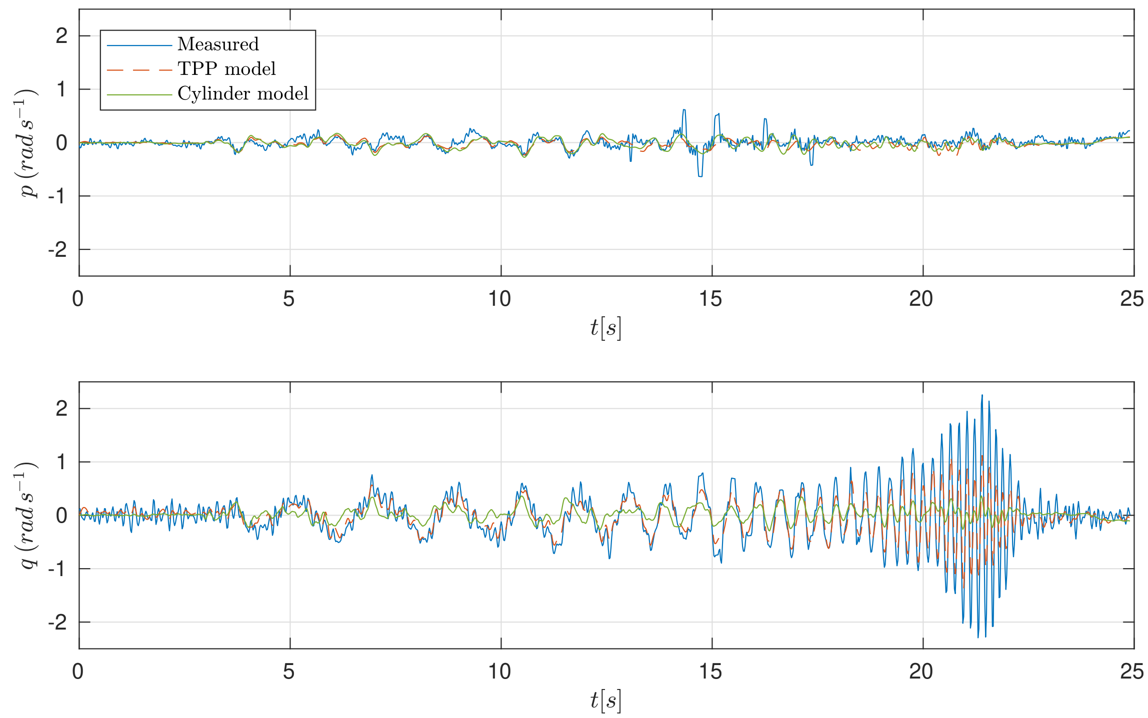


Figure 9: Validation chirp on the pitch axis in forward flight. The CD model's accuracy is much worse than the TPP model, while the latter is also unable to follow the measurement signal at higher frequencies. At around $15\,\mathrm{s}$, the logging system shows some delays, leading to dropped measurements.

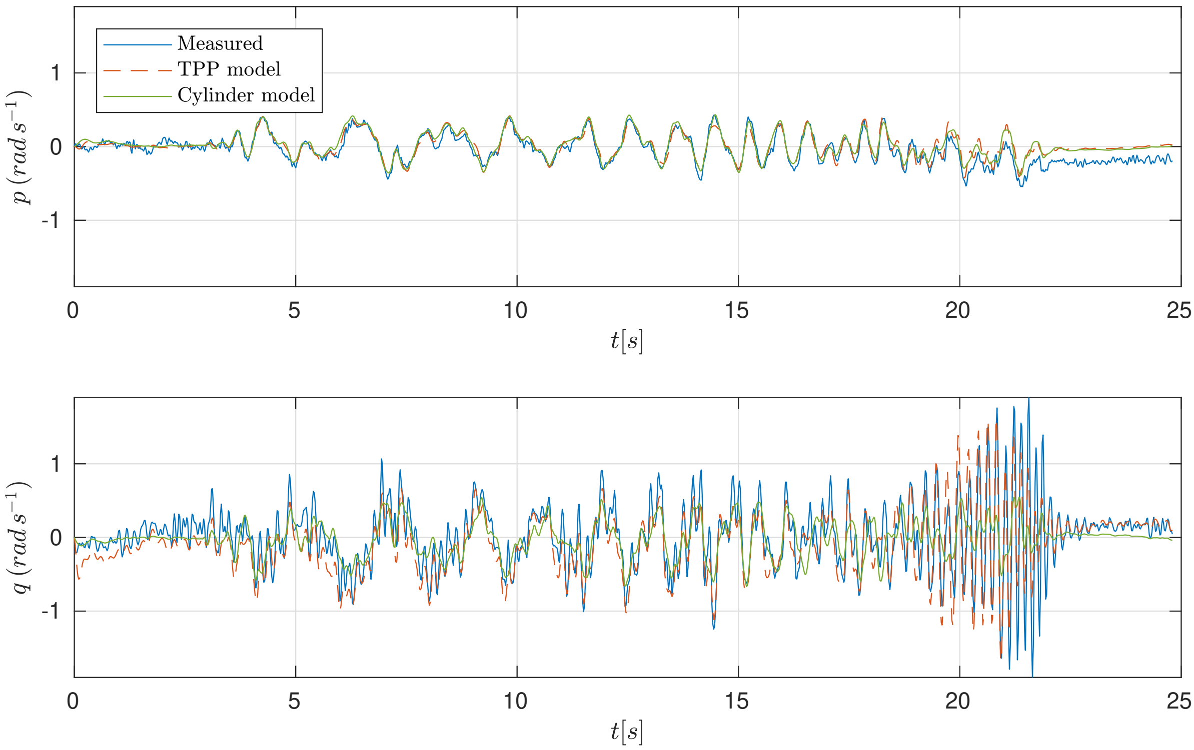


Figure 10: Validation chirp on the elevator axis in forward flight. The roll response q (yaw in standard aircraft reference frame) is fitted accurately by both TPP and CD models, while from around $15\,\mathrm{s}$ the CD model is unable to replicate the higher frequencies that the TPP model can still generate. The TPP model is still unable to replicate the highest frequencies of the chirp.

DelftaCopter Propulsion Optimization from Hover to Fast Forward Flight using Windtunnel Measurements

C. De Wagter*, B.D.W. Remes†, R. Ruijsink, E. van der Horst, F. van Tienen, D. van Wijngaarden , J.F. Meulenbeld, K. van Hecke

Abstract

Enlarging the flight envelope of aircraft has been a goal since the beginning of aviation. But requirements to fly very fast and to hover are conflicting. During the design of the **D**elfta**C**opter, a tail-sitter hybrid UAV with a single large rotor for lift in hover and propulsion in forward flight, the design of the rotor needs to properly balance hovering requirements and fast forward flight requirements. The initial design with a one meter rotor placed too much emphasis on efficiency in hover, while most flights consist of very short periods of hover and very long phases of forward flight. Two new rotor designs and corresponding motors were tested an open jet wind tunnel. The propulsion system was tested from hover conditions to very fast forward flight in search of the most optimal operating point for each condition. The resulting system requires merely more power than the initial rotor in hover while it is capable of much faster forward speeds. The power requirements are shown to be compatible with modern power sources like Lithium-Ion batteries, which form the next step in improving the efficiency of hover-capable fast UAV.

1 Introduction

Extending the flight endurance and flight range of aircraft has been a goal since the beginning of aviation. This has typically been solved by increasing the size of aircraft to carry more fuel. But as Unmanned Aerial Vehicle (UAV) were gaining in popularity, this has re-triggered the quest for small and efficient platforms. Many real-life applications have a combined need for long range but also vertical take off and landing [1, 2]. Unfortunately these requirements are conflicting.

Hybrid UAV have been proposed to address the combined needs of long range and hovering capability [3]. By using a hovering set of rotors, vertical take off and landing capability was added to an efficient fixed wing airframe, which enables long range flights [4].

To further optimize the efficiency, a single rotor is more efficient than several smaller rotors. The **D**elfta**C**opter[1] is a platform that uses this approach. Using collective pitch, the rotor can be reconfigured for optimal hover and optimal fast forward flight. Nevertheless, finding the combined optimum of hover and forward flight remains a challenge as for hover an as large as possible rotor would be desired for efficiency, while for forward flight at high speeds, a much smaller propeller is optimal [5]. To assess the efficiency, several rotors are tested in windtunnel and subsequently in forward flight.

Section 2 presents the windtunnel measurements. Section 3 gives the results of the outdoor test flights. Finally, Section 4 gives the conclusions.

2 Windtunnel

Figure 1: The new **D**elfta**C**opter Propulsion System is mounted on a static test rig in the TUDelft Open Jet windtunnel. The test setup includes force measurements, moment measurements, voltage, current, airspeed, rotor pitch, throttle and rotor rpm measurements.

Windtunnel measurements were performed in the TUDelft open jet windtunnel. A rotor system was mounted on a static rig in front of the opening as shown in Figure 1. The rotor was placed on a RC-Benchmark Series 1780 force

*Email address: c.dewagter@tudelft.nl Delft University of Technology, Kluyverweg 1, 2629HS Delft, the Netherlands

†Email address: B.D.W.Remes@tudelft.nl

[1] `http://www.delftacopter.nl/`

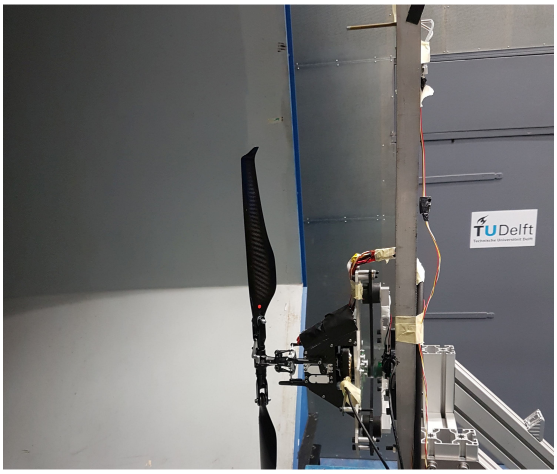


Figure 2: Close up of the new rotor blades (left), the force and moment balance (middle) and the airspeed probe (on top). The windtunnel blows from left to right.

and moment measuring device[2]. The balance not only logged forces and moments but also logged total current consumption, voltage and Rotations Per Minute (RPM). A pitot tube was recording the local air speed, which can be seen in Figure 2. Onboard measurements were performed onboard a Paparazzi-UAV [6] autopilot board. The measurements included RPM, Voltage, Airspeed, Current reported by the Electronic Speed Controller (ESC), Throttle commands and Collective pitch commands. Two separate logfiles were obtained, namely one from the balance and one from the autopilot. The logs were then synchronized by aligning the measurements that were obtained by both, namely the RPM and the current.

The autopilot was programmed to systematically step through its entire pitch and throttle range as illustrated in Figure 10. The whole process was repeated for two sets of rotor blades, namely the $24 inch$ and $26 inch$ blades from *T-Motor*. Several combinations of airspeed, throttle and pitch lead to destructive combinations, being it either due to over-RPM, over-current, over-temperature or any RPM that would make the setup vibrate excessively. Therefore, the range of pitch and throttle values were manually limited to safe conditions. For every pitch, throttle, rotor and airspeed combination, the autopilot would wait 3 seconds for the RPM, Current and flow to stabilize. An automated analysis tool in *MATLAB* then averaged the values during the steady phase only, which is shown as red crosses in Figure 10.

Figures 11 show the obtained net thrust for various throttle and collective pitch settings and various airspeeds. The required power to obtain this thrust is shown in Figure 12. Finally, Figure 13 shows an estimation of the obtained efficiency[3].

3 Test Flights

3.1 Power in function or RPM

Figure 3: Take-off of the **D**elfta**C**opter PH-3MM during an outdoor test-flight.

To validate the figures found in the wind tunnel tests, outdoor test flights are performed. A **D**elfta**C**opter was registered under the Dutch CAA-NL as *PH-3MM* and is shown in Figure 3. The UAV was flown at a variety of throttle levels and collective pitch values, and the resulting airspeed and power are then used to find the optimum.

Figure 4 shows the decreasing RPM as the collective pitch is increased and the throttle decreased while the airspeed is kept relatively constant. Figure 5 shows the relation between throttle, collective pitch and rotor RPM that leads to a constant airspeed of about 22 m/s.

The required power to fly a this airspeed depends on rotor RPM and is shown in Figure 6. It can clearly be seen that lower RPM are more efficient as the power used ($P = U \cdot I$) of lower to fly at the same airspeed ($P = V \cdot Drag$).

3.2 Power in function of speed

A second test flight was performed at varying airspeed. The power required in function of the airspeed is shown in Figure 7. The third power fit ($P = f(V^3)$) is shown in red.

The track that was flown is shown in Figure 8. Notice the increasingly large turn radius as the airspeed increases while the **D**elfta**C**opter makes turns with a limited bank angle that is maxed out during most of the turn.

The raw airspeed and current in function of time is given in Figure 9. In the time frame from 20 to 30 minutes into the flight, the speed was increased. The hovering phases are

[2] Max thrust: 25 kg, max torque 12Nm, max voltage 60V, max current 100A continuous and 150 burst.

[3] This value is highly influenced by the accuracy of the current and force calibrations.

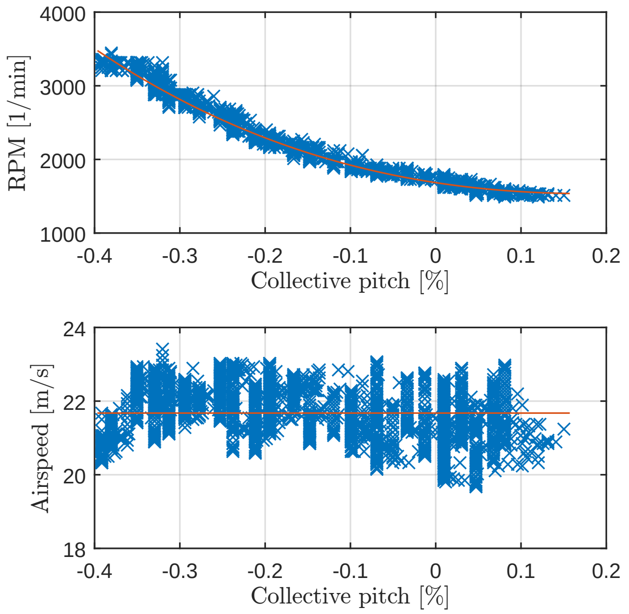


Figure 4: RPM in function of collective in forward speed for a constant airspeed.

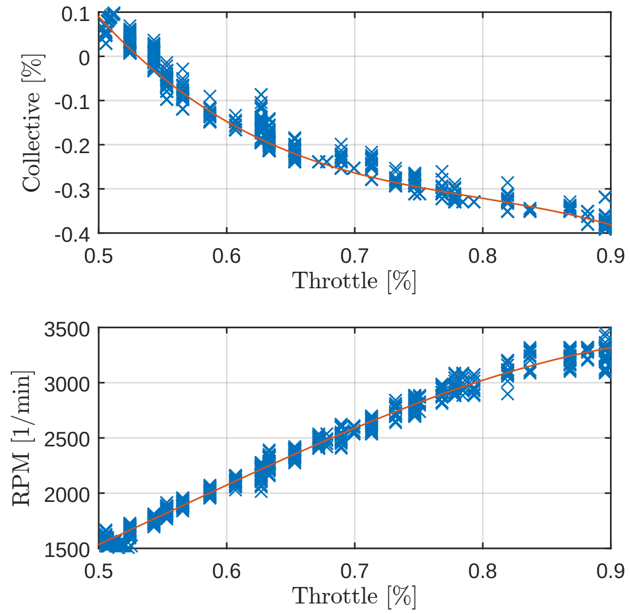


Figure 5: RPM in function of collective in forward speed for a constant airspeed.

clearly recognizable as the airspeed drops to zero while the used current increases to over $25Amp$.

3.3 Trade-off between 24 *inch and* 26 *inch rotor*

A significant difference in efficiency in forward flight could not be found at speeds below 25 m/s. During hover, however, a very significant difference was observed. The motor and rotor combinations can still hover with much lower battery voltages. To increase the range of the **Delfta**Copter, Lithium-Ion batteries are used that provide low discharge rate and show significant voltage drops when loaded at the limit.

With the 24 inch ($61cm$) rotor, **Delfta**Copter could only empty its battery 70% while still being able to hover. Using the larger 26 inch ($66cm$) rotor, **Delfta**Copter could empty its battery to 90% before the voltage drop would make it impossible to hover. This is due to the higher voltage drop of the battery by the higher load of the less efficient smaller rotor on the one hand, and because of the higher voltage needed by the motor to reach a higher RPM on the other hand. This significant difference of 20% was deemed more important than the slight increase in forward flight efficiency.

3.4 Comparison with 2016 rotor design

The $1m$ diameter rotor 2016 **Delfta**Copter could hover using significantly less power than the new $66cm$ (26 inch) rotor. But in forward flight however, the opposite is true. Since **Delfta**Copter spends way more time in forward flight than in hover, overall the smaller rotor yields a huge boost in range.

The smaller rotor and motor with more torque also has indirect advantages. Upon stall for instance, the rotor picks up RPM much faster when switching to hover mode. This allows the new **Delfta**Copter to recover from much more dramatic situations.

The smaller rotor also has complications. The control is further away from helicopter control. This is the topic of a different study.

4 Conclusion

A new propulsion design for the **Delfta**Copter was tested in the Open Jet Windtunnel Facility (OJF) wind tunnel and subsequently in real test flights. The wind tunnel measurements have shown that the rotor can be efficient over a wide

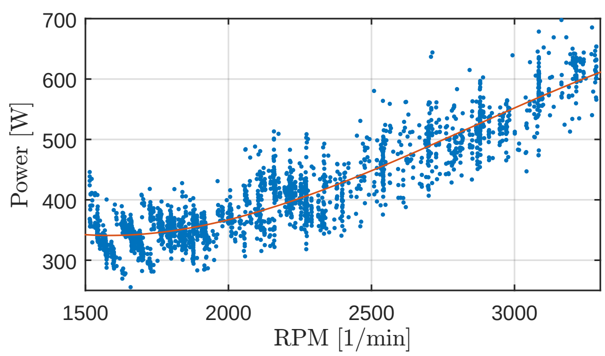


Figure 6: Power required to fly a 22 m/s at various RPM.

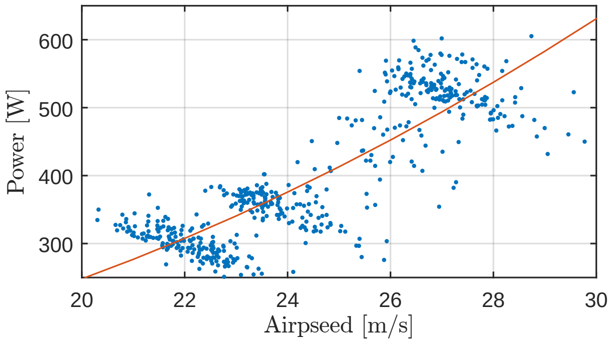


Figure 7: Power versus airspeed during outdoor test flight.

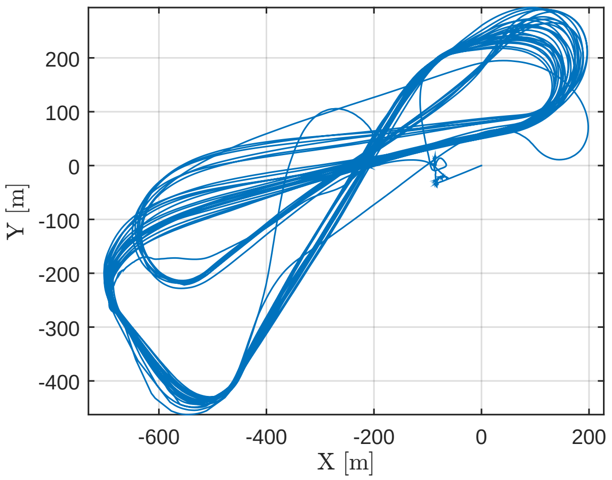


Figure 8: Top view of the track of the 50 minute test flight.

range of RPM. The optimal RPM for a given situation could be obtained and subsequently used in outdoor test flying. Since no accurate drag of the fuselage was measured in the windtunnel, real performance data was obtained from outdoor testing in real world conditions. While noise levels in the outdoor measurements are high, nevertheless, accurate performance data was obtained. The new 2018 **D**elfta**C**opter rotor and motor performance is compared with the 2016 **D**elfta**C**opter rotor design. The efficiency at high speed is shown to be dramatically improved, while hovering capabilities are not compromised. Overall the capabilities of the **D**elfta**C**opter were highly improved.

ACKNOWLEDGMENTS

The authors would like to thank the 2018 Outback Medical Challenge Organization to provide a challenging real-world environment to push the capabilities of UAV.

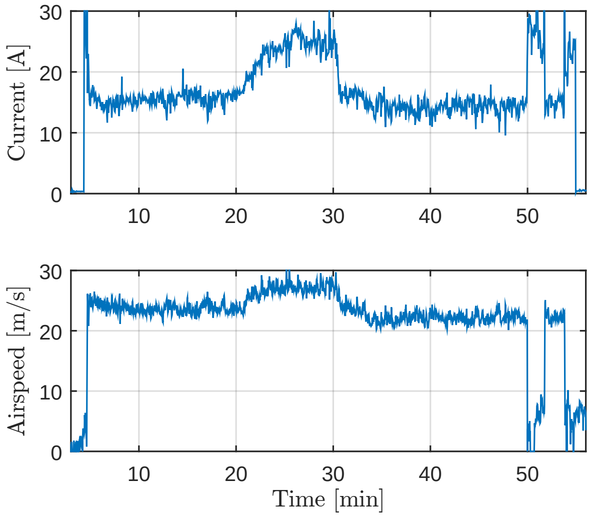


Figure 9: Used current of the 6-cell Li-Ion battery.

	2016	2018
Rotor diameter	100 cm	66 cm
Power in hover	≈ 600 Watt	≈ 700 Watt
Power in forward flight at 20 ms	≈ 400 Watt	≈ 260 Watt
Maximum forward speed	≈ 21 m/s	≈ 28 m/s

Table 1: Comparison between the 2016 and 2018 **D**elfta**C**opter.

REFERENCES

[1] Francesco Nex and Fabio Remondino. Uav for 3d mapping applications: a review. *Applied geomatics*, 6(1):1–15, 2014.

[2] Geraldo José Adabo. Long range unmanned aircraft system for power line inspection of brazilian electrical system. *Journal of Energy and Power Engineering*, 8(2), 2014.

[3] S Herbst, G Wortmann, and M Hornung. Conceptual design studies of vertical takeoff and landing remotely piloted aircraft systems for hybrid missions. *CEAS Aeronautical Journal*, 7:135–148, 2016.

[4] C. De Wagter, R. Ruijsink, E.J.J. Smeur, K. van Hecke, F. van Tienen, E. v.d. Horst, and B. Remes. Design, control and visual navigation of the delftacopter. *Journal of Field Robotics*, pages 1–24, 2018.

[5] Quentin R Wald. The aerodynamics of propellers. *Progress in Aerospace Sciences*, 42(2):85–128, 2006.

[6] Gautier Hattenberger, Murat Bronz, and Michel Gorraz. Using the paparazzi uav system for scientific research.

In G. de Croon, E.J. van Kampen, C. De Wagter, and C. de Visser, editors, *IMAV 2014, International Micro Air Vechicle Competition and Conference 2014*, pages 247–252, Delft, The Netherlands, aug 2014.

APPENDIX A: PLOTS

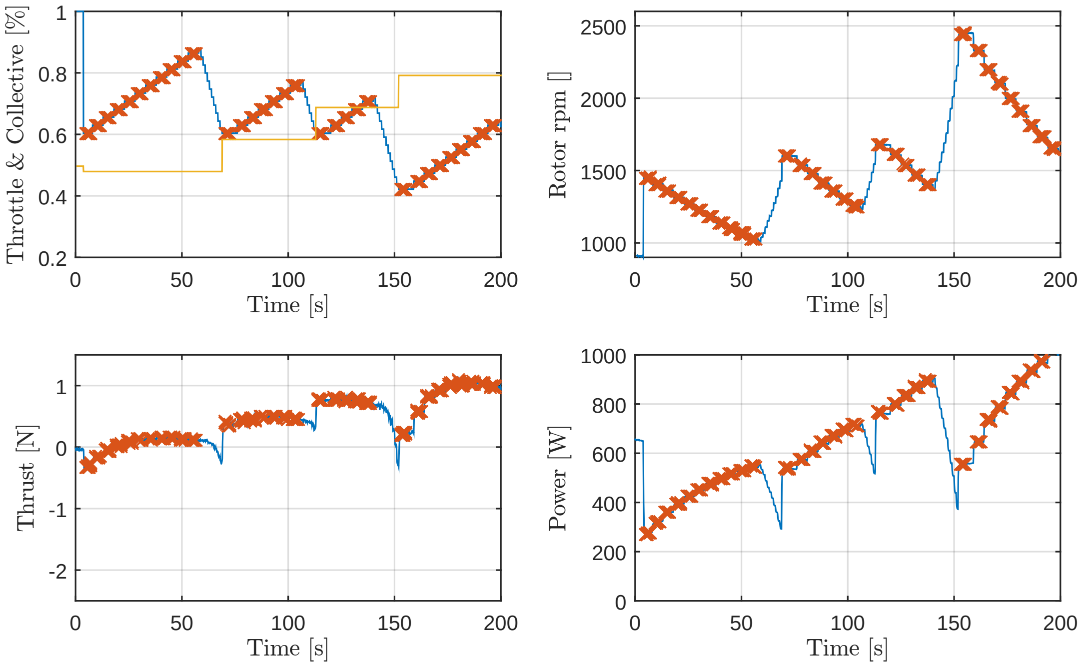


Figure 10: A fragment of a time series obtained from the OJF wind tunnel testing. The autopilot commands an series of pitch and throttle settings for 3 seconds each. Once everything is stabilized, the average over 1.5 seconds of measurement is taken (Red X).

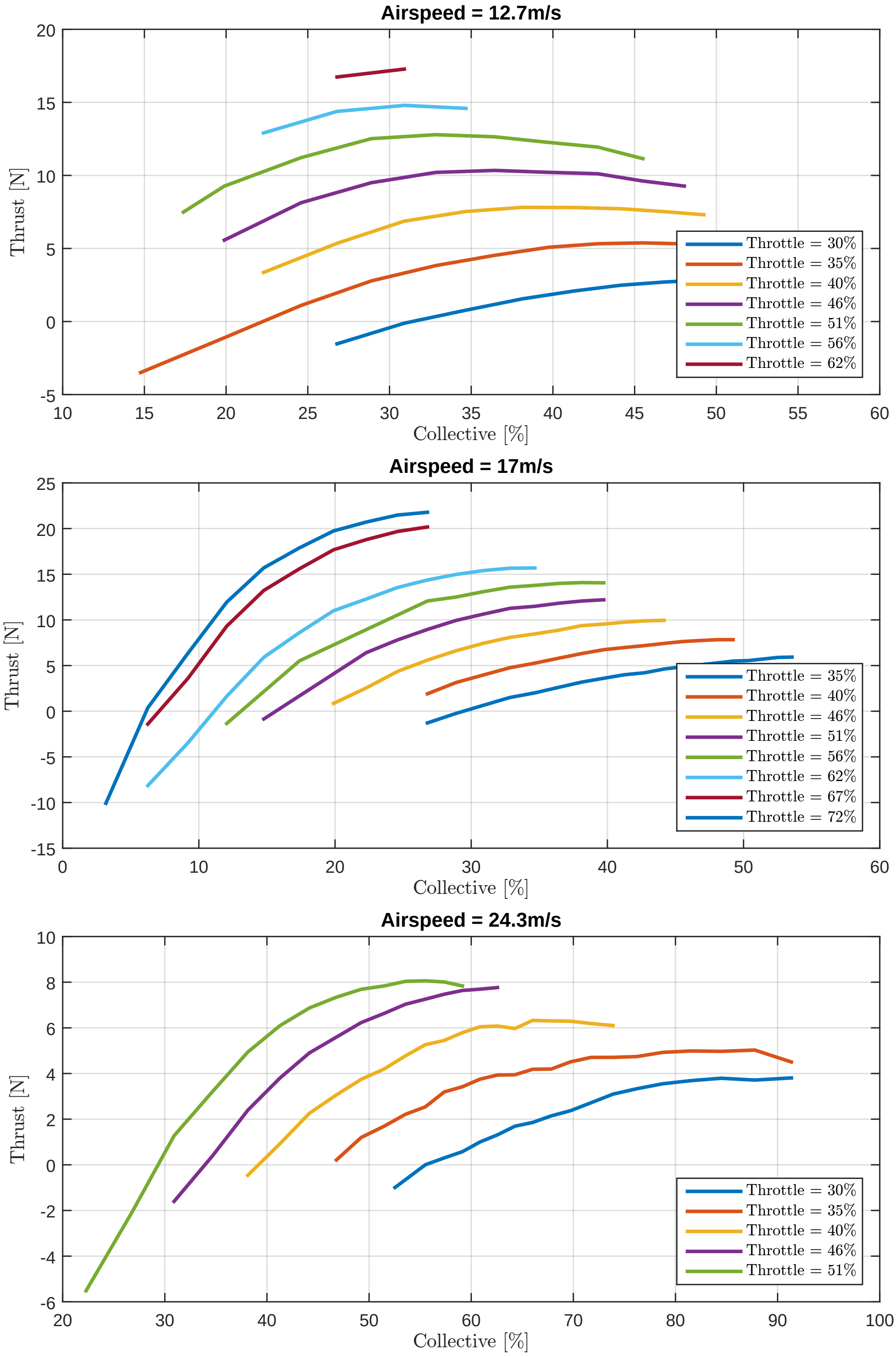


Figure 11: Propulsion power in function of thrust for the 24 inch **D**elfta**C**opter rotor.

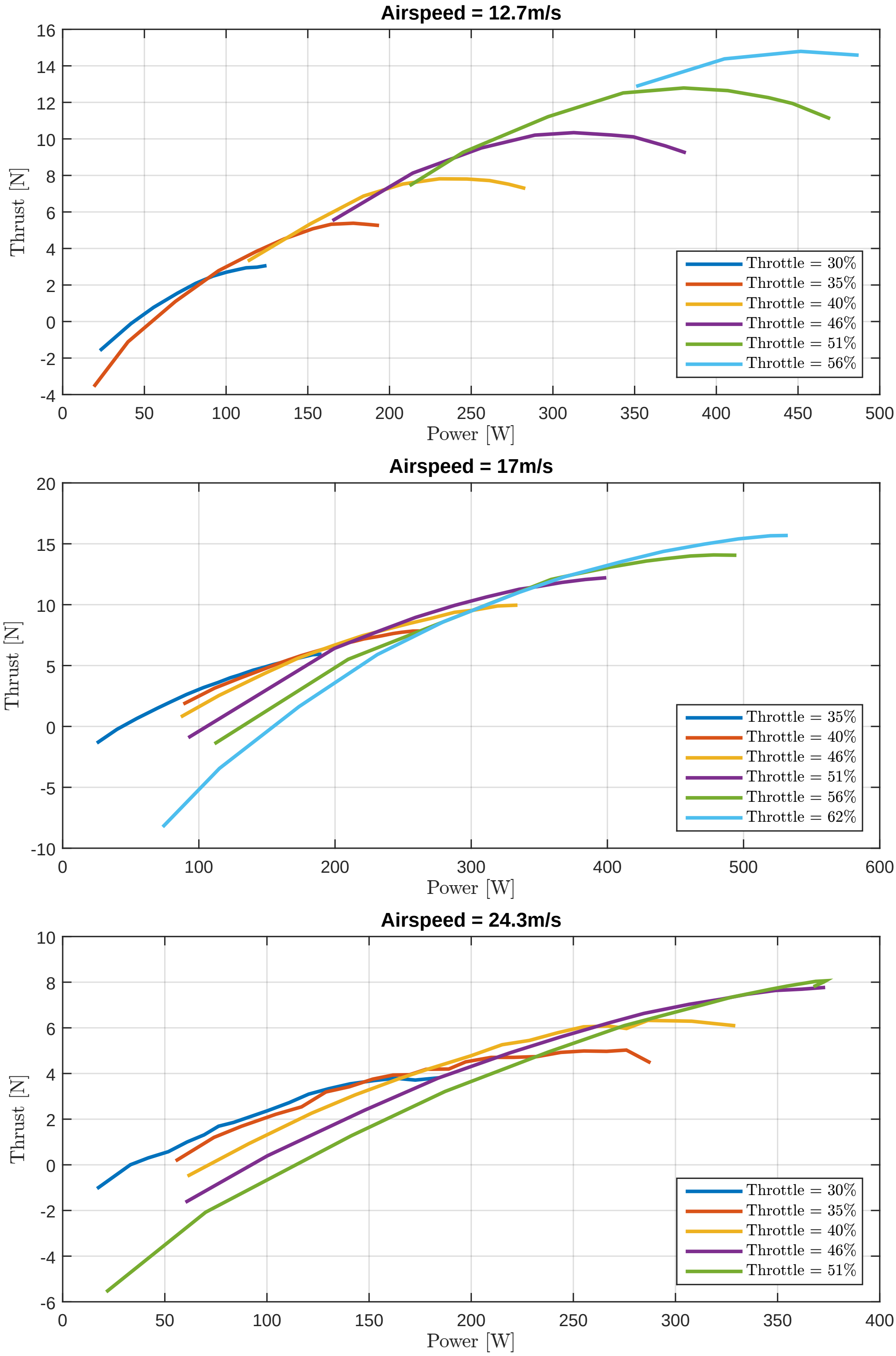


Figure 12: Power in function of Thrust for the 24 inch **D**elfta**C**opter rotor.

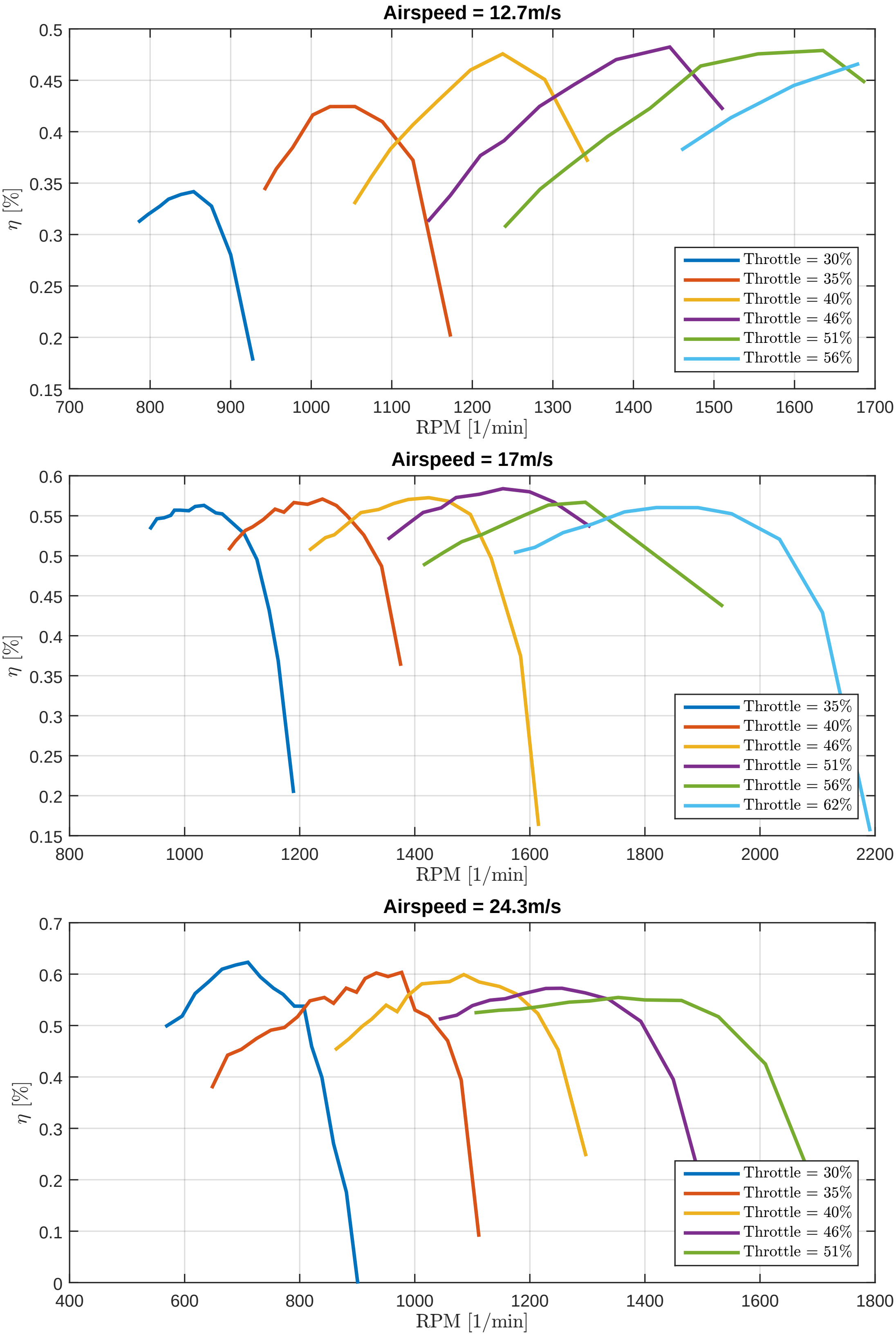


Figure 13: Propulsion efficiency in function of rotor speed for the 24 inch **Delfta**Copter rotor.

Increased autonomy for UAV using solar panels and take advantage of updrafts through algorithm

Leandro D. de Jesus, Alexandre R. Brandão, Caique de S. F. Duarte and Felix Mora-Camino
Federal University of Itajuba (UNIFEI), Brazil
Fluminense Federal University (UFF), Brazil

ABSTRACT

The purpose of this research is to join more than one research and create an electric UAV that uses solar plates to recharge the battery and explores the benefits of the upward winds to be able to make autonomous flights of long distances, such as border monitoring, among other types of flights. In this way, we can observe that with the improvement of these two techniques we can gain a very large autonomy increase, increasing the flight coverage distance, transforming the electric UAV into a very viable alternative for some types of service, managing to eliminate its biggest problem which would be the time of discharge of the battery.

1 INTRODUCTION

Unmanned aerial aircraft are increasingly being used for various purposes such as forest monitoring, locating endangered people or objects in a large area.

For countries where there is a vast border airport where it is difficult to monitor even manned aircraft where it ends up being a vey high price, due to the fact of the fuel, maintenance of the aircraft, then there are emerging alternatives of using unmanned electric aircraft for being a cheaper solution and letting manned aircraft be used in more emergency locations.

However it still has a very big problem because the autonomy of the batteries used by these UAVs are of vey short duration, having to be forced at some time to land them to replace these batteries and continue their mission. There are many works being done to lower the energy consumption of the aircraft's embedded system, as well as altering its design [1,2,3,4,5], but the focus of this work is to use the researched ones carried out on flights with eletric aircraft using solar cells in order to recharge the battery, where for Brazil that is a tropical country we can have many benefits of this technology and use the research done on upwinds that can be much used to take the expeses of the engines, letting only the air current keep the aircraft flying.

In the next Chapter will be presented some historical projects on flights of aircraft that used solar plates [6], then will be presented historical facts about the use of rising winds [7] and to finish I will talk about the experiment that will be carried out.

2 HISTORY OF SOLAR-POWERED FLIGHT

I will present a summary of important projects involving unmanned aircraft using solar panels.

2.1 The SoLong

There are many projects carried out using solar panels to increase flight durability such as the SoLong project that was made by AC Propulsion Inc., a company that specializes in high efficiency electric propulsion. Alan Cocconi who is the founding chairman and chief engineer of AC Propulsion has carried ou project financing. SoLong's goal was to be able to demonstrate that the aircraft could fly for several days powered by

leandrodinizjesus@hotmail.com

Increased autonomy for UAV using solar panels and take advantage of updrafts through algorithm

Leandro D. de Jesus, Alexandre R. Brandão, Caique de S. F. Duarte and Felix Mora-Camino
Federal University of Itajuba (UNIFEI), Brazil
Fluminense Federal University (UFF), Brazil

solar energy. SoLong was a solar powered monoplane with a wingspan of 4,75m, a wing area of 1.5 m^2, with a battery that weights 5.6kg [Sanyo 18650 lithium-ion battery with 220 Wh / kg] 76 SunPower A300 solar cells and a total mass of 12.6kg. The plane was controlled remotely by six experienced pilots who concentrated on trying

2.2 The Helios platform (Heliplat)

Heliplat was developed to fly in the stratosphere, where it was the first unmanned aerial vehicle (UAV) with this capability and long duration in Europe. The monoplane had eight brushless motors, a double boom tail and two rudders [10]. The Heliplat was designed to fly at altitudes over 17km up to 25km, aiming to provide information assistance services in parts of the Mediterranean Sea area. The project was an offshoot of the HELINET project, where his objective was to create a network of stratospheric platforms to carry out traffic monitoring, environmental surveillance and broadband services coordinated by Politecnico di Torino. In January 2000, the project was funded by the European Commission under the Fifth Framework Program and developed at the Polytechnic University of Turin. The goal was to create a UAV that would have resistance to flights at high altitudes using solar panels and fuel cells that could extend flight durability in up to approximately 9 months. But this project was not finalized by limited financial support, a scale-sized solar-powered prototype was fabricated from it [11, 12, 13, 14].

2.2 The Sky-sailor

to use upward currents and avoided downward currents. SoLong was successful in making its flight at Desert Center Airport on June 3, 2005, which is located east of the Colorado desert in California, establishing a 48-hour flight without landing. They did not continue the flight test any longer because the pilots were exhausted [8,9].

Sky-sailor was a project created by Space Technology Advances by Resourceful, Targeted and Innovative Groups of Experts and Researchers of the European Space Agency (ESA).

The project began at the end of 2003, aiming to get a very light aircraft that uses solar panels to fly night and day. The research was conducted by the Autonomous Institute of Systems Laboratory of EPFL. The Sky-sailor's overall configuration was similar to that of a motorized glider, with a base layout similar to Glider Abance, where it had won world records, distance and duration. The aircraft had a wingspan of 3.2m, a lithium-ion battery that weighs 1056kg and a total weight of 2444kg. His first flight was held in 2005.

In June 2008, a demonstration flight was held where it lasted 27 hours, setting a record [15].

2.3 The Venture

In 2007, the US Defense Advanced Research Projects Agency (DARPA) was looking for an alternative to expensive satellites. The purpose of the alternative was to meet ISR requirements or to relay communication. With it a project named Vulture, which was a solar powered aircraft that

Increased autonomy for UAV using solar panels and take advantage of updrafts through algorithm

Leandro D. de Jesus, Alexandre R. Brandão, Caique de S. F. Duarte and Felix Mora-Camino
Federal University of Itajuba (UNIFEI), Brazil
Fluminense Federal University (UFF), Brazil

could stay in the air for up to five years. The Vulture weighed 450kg and had a payload of 5kW. In 2009 the project entered a second phase, with the aim of being an unmanned aerial system of high altitude and long duration that could remain for three months flying. After a competition for three team to build the plane the winner was the SolarEagle, with an extended version of Zephyr and made its first flight in 2014 [15, 16].

3 HISTORICAL FACTS OF AIRCRAFTS USING UPWARD WINDS

First studies involving upward winds were conducted through bird flight studies [17], where winds are coming from the ground upwards and greatly assist the birds to glide as for UAVs to save battery during flight. The benefits from these winds are that the aircraft manages to increase its resistance and to save energy. A study conducted by NASA using theoretical calculations demonstrates that a 2 hour resistance UAV can achieve a flight of maximum 14 hours using updrafts in good weather conditions [18].

I will present some studies on thermo exploitation strategy using UAV.

The first to propose an autonomous simulation for the UAV was John Wharington in 1998 [19].

In this work an updraft modeling structure for simulations of increasing UAV strategies was presented. He used a simplified thermal model based on measurements and theoretical calculations. In the model presented the thermal is considered a circle or ellipsoid, with a distribution of Gaussian vertical velocity. In other simulations performed later, others ended up using very similar thermal vertical velocity distributions, in a quadratic function [20] or ray dependent [21].

First he used a strategy that is well known among glider pilots, based on the rules of the famous racing glider pilot, Helmut Reichmann [22]. The rules are:

If the climb is good, lower the seat angle.

If the climb improves, increase the angle of the seat.

If the rise remains constant, keep the bank angle constant.

This strategy worked very well in simulations, but the method used as previously mentioned was a simplified good where it would have other methods more effective than it. With the theoretical calculation already solved he developed an orientation algorithm using reinforcement learning and a neural network system that located the center of a thermal for an optimal autonomous exploration of the thermals. But his algorithm with neural network-based thermal center locator consumed a lot of time for real-time online applications.

Stephane Doncieux and his colleagues developed a growing strategy using evolutionary algorithm to optimize the connection weights in a neural network [23], where altitude gain during the allocated evaluation period was used as fitness function. The input parameters of the neural network were the vertical speed, the angle of rotation and slope of the glider. The exits of the

Increased autonomy for UAV using solar panels and take advantage of updrafts through algorithm

Leandro D. de Jesus, Alexandre R. Brandão, Caique de S. F. Duarte and Felix Mora-Camino
Federal University of Itajuba (UNIFEI), Brazil
Fluminense Federal University (UFF), Brazil

network controlled the elevator and the rudder of the airplane. Using simulations, this strategy was successful for ideal thermals, modeled similar to those proposed by Wharington [19].

NASA during the Autonomous Ascent Project at Dryden Flight Research Center developed the first successful UAV [24]. Michael Allen and a team of engineers programmed a small UAV (4.27 m, 6.8 kg glider called Cloud Swift) to detect if it is in an upward current and use that circling current. During the project, the UAV flew 17 times, gained an average altitude of 173m in 23 updrafts and climbed 844m in a strong thermal. In one of the flights of this project, the UAV added 60 minutes to its resistance rising autonomously.

More recently, even more successful location and thermal orientation algorithms were designed and implemented at the State University of North Carolina and the US Naval Research Laboratory by Daniel J. Edwards [25]. The UAV in this project participated in the Valley Valley Cal Race in May 2009, where he defeated the humans in a cross-country competition and covered over 113km. On another flight remained in the air for more than 5.3 hours.

4 EXPERIMENT

We will use for this experiment a model airplane known as wing-zags with a wingspan of 2.25 meters, has a angle of 30 degrees, ZAGUI12 wing profile, done on styrofoam type 5 (P3). Figure 1 shows the UAV previously referenced already with the embedded system, but without the solar plates.

Figure 1 - UAV that will be used in the experiment without solar plate

An amount of 20 solar cells will be allocated in the wing where on the one hand will have on average 10 cells connected in series and each wing will be connected in parallel. The solar panels are made of monocrystalline silicon material, measuring 63 x 125 mm, which generates 0.574 volts, generate a chain 2,915 amperes, with a 21.8% efficiency.

For the autonomous flight will be used a pixhawk controller board where with it will be a zybo zynq 7000 type computer that has 650 MHz dual-core ARM Cortex-A9 processor, DDR3 memory and programmable logic equivalent to FPGA Artix-7, where it will be running Ubuntu 16.04 with the ROS program to do optimization of performance in a thermal. The algorithm for optimization will be implemented in python.

The purpose is to generate a mission autonomously for aircraft and to measure the time that it has been accomplished this mission to obtain results in the part of autonomy.

5 CONCLUSION

We can conclude with this paper that many works of UAV using solar panels and rising winds gave a very significant increase of autonomy. With this information, its being

Increased autonomy for UAV using solar panels and take advantage of updrafts through algorithm

Leandro D. de Jesus, Alexandre R. Brandão, Caique de S. F. Duarte and Felix Mora-Camino
Federal University of Itajuba (UNIFEI), Brazil
Fluminense Federal University (UFF), Brazil

built an eletric UAV with solar panel coupled in the fuselage, an algorithm is being implemented so that the UAV can autonomously identify that it is entering a rising wind and take advantage of this benefit to save battery power with the engine.

6 FUTURE WORKS

It will be creating an electric circuit for charging the battery efficiently and the code will be developed in Python where when entering a thermal the motor shuts off until it has passed the terminal to have a consumption of battery assisting in moments that the solar plate does not generate enough power to charge the battery.

REFERENCES

[1] Serokhvostov S.V., Churkina and T.E. Optimal Control for the Sun-Powered Airplane in a Multi-Day Mission. Scientific Proceedings of Riga Technical University, Series 6 "Transport and Engineering. Transport. Aviation Transport. 27: 212–220. 2008.

[2] Reza Malekian ,A. F. Kavishe, B. T. Maharaj, P. K. Gupta, G. Singh and H. Waschefort. Smart Vehicle Navigation System Using Hidden Markov Model and RFID Technology. Springer Science+Business Media New York, 2016.

[3] Alexandr V. Kornushenko, Andrey V. Shustov, Sergey V. Lyapunov and Sergey V. Serokhvostov. Numerical, Experimental and Flight Investigation of MAV Aerodynamics. 3rd US-European Competition and Workshop on Micro Air Vehicle Systems (MAV07) & European Micro Air Vehicle Conference and Flight Competition (EMAV2007), 17-21, Toulouse, France. 2007.

[4] Serokhvostov S.V. Maximization of horizontal flight time for airplane with electrical power plant. Research Bulleting of Institute of Aeronautics and Applied Mechanics, N12, Warsaw, 131-133. 2001.

[5] P. K. Gupta and G. Singh. A Novel Human Computer Interaction Aware Algorithm to Minimize Energy Consumption. Springer Science+Business Media New York. Wireless Pers Commun (2015) 81:661–683, 10.1007/s11277-014-2151-y. 2014.

[6] Xiongfeng Zhu, Zheng Guo and Zhongxi Hou. Solar-powered airplanes: A historical perspective and future challenges. Progress in Aerospace Sciences, Volume 71:36-53. 2014.

[7] Zsuzsa Ákos, Máté Nagy, Severin Leven and Tamás Vicsek. Thermal soaring flight of birds and UAVs. Bioinspir. Biomim., 5, 045003. 10.1088/1748-3182/5/4/045003. 2010.

[8] AC propulsion SoLong UAV flies for 48 h on sunlight two nights aloft opens new era of sustainable flight. AC Propulsion report; 2005.

[9] AC propulsion's solar electric powered SoLong UAV. AC Propulsion report; 2005.

Increased autonomy for UAV using solar panels and take advantage of updrafts through algorithm

Leandro D. de Jesus, Alexandre R. Brandão, Caique de S. F. Duarte and Felix Mora-Camino
Federal University of Itajuba (UNIFEI), Brazil
Fluminense Federal University (UFF), Brazil

[10] Romeo G, Frulla G, Cestino E and Corsino G. HELIPLAT: design, aerodynamic, structural analysis of long-endurance solar-powered stratospheric platform. Jaircr .41(6), 1505–20. 2004.

[11] Frulla G and Cestino E. Design, manufacturing and testing of a HALE-UAV structural demonstrator. Compos Struct,83(2), 143–53. 2008.

[12] Romeoa G, Pacinoa M and Borello F. First flight of scaled electric solar powered UAV for mediterranean sea border surveillance forest and fire monitoring. JAerosp Sci Technol Syst, 1(88), 8–19. 2009.

[13] Cestino E. Design of solar high altitude long endurance aircraft for multi payload and operations. Aerosp Sci Technol, 10(6), 541–50. 2006.

[14] Noth A, Engel MW and Siegwart R. Flying Solo and Solar to Mars. IEEE Robot Autom Mag, 13(3), 44–52. 2006.

[15] Marsh G. Best endurance under the sun. Renew Energy Focus, 11(5), 24–7. 2010.

[16] Broad Agency Announcement (BAA) Vulture II. DARPA-BAA-10–04, 2009.

[17] L. Rayleigh. “The soaring of birds,” Nature, 27, 534–535, 1883.

[18] Allen M J. Autonomous Soaring for Improved Endurance of a Small Uninhabited Air Vehicle. AIAA-2005-1025. 2005.

[19] Wharington J. Autonomous Control of Soaring Aircraft by Reinforcement Learning Doctorate theses, Royal Melbourne Institute of Technology, Melbourne, Australia. 1998.

[20] Fonseka E. Modeling and Flying Thermal Tubes with an UAV Semester Thesis Report, Autonomous System Laboratory, ETH Zürich, Switzerland. 2007

[21] Allen M J. Updraft model for development of autonomous soaring uninhabited air vehicles. AIAA. 2006.

[22] Reichmann H. Cross-Country Soaring. Thomson Publications, Santa Monica, CA. 1988.

[23] Doncieux S, Mouret J B and Meyer J-A. Soaring behaviors in UAVs : 'animat' design methodology and current results. In 7th European Micro Air Vehicle Conference (MAV07), Toulouse. 2007.

[24] Allen M J. Guidance and Control of an Autonomous Soaring UAV. AIAA-2007-867. 2007.

[25] Edwards D J. Implementation Details and Flight Test Results of an Autonomous Soaring Controller North Carolina State University, Raleigh, NC, 27604. 2008.

On the Response of Leading-Edge Phenomena and Near-Wake Formations to Trailing-Edge Flap Actuation

Albert Medina* and Matthew Rockwood
Air Force Research Laboratory, Wright-Patterson AFB, OH 45433

Abstract

In this work, the response of a separated flow field and free shear layer of a NACA 0006 wing to rapid mechanical actuation of a conventional trailing-edge flap is studied in a nominally two-dimensional flow at a chord-based Reynolds number of 4×10^4. We report on the transient lift response to rapid step-and-hold maneuvers and the response of the near-body wake to sustained harmonic pitching of the flap. The term "rapid" is reserved here for motions performed within one convective time. The wing is fixed at $\alpha = 10°$ and $20°$ for representative separated and massively-separated flow fields, respectively. In both kinematic cases the amplitude of flap motion is low, measuring $1°$. It is shown that the rapid motion of the trailing-edge flap is sufficient to induce flow reattachment or rollup of the separated shear layer into a leading-edge vortex. Under harmonic flap excitation, it is shown that authority of near-body wake shedding characteristics may be imposed even for the separated transitional case of $\alpha = 10°$.

1 Introduction

The mitigation, control, and exploitation of separated flows ranks with turbulence as one of the great unresolved problems in aerodynamics. Principal applications pertain to small flight articles where skin friction drag is not the primary driver of aerodynamic performance. Such small flight articles are renowned for their agility and maneuverability, often allowing the article to perform large excursions momentarily from steady cruise flight conditions. Due to the scale, however, small flight articles may be rendered particularly susceptible to relatively large disturbances in the flow field. In both instances, be it the article's high maneuverability or incurred flow disturbances, the article is subjected to temporal variations in aerodynamic loading accompanied by transient vortical phenomena. If left unabated, the asymptotic state of transient flows is outright separation of the lifting surface. In such instances of extensive separation, the question arises as to whether it is preferable to reduce separation to attain the putatively higher aerodynamic efficiency, or whether it is preferable to manage the separated flow to produce a favorable aerodynamic effect, such as higher lift. To this end, the proper characterization of the dynamic response of the flow to actuator input is imperative for the development of effective active flow control solutions. In this study we propose the use of mechanical actuation of the trailing-edge flap to induce a transient response in aerodynamic loading from rapid flap-step input and explore the effects on near-body shedding dynamics from sustained harmonic flap motion.

Conventional active flow control techniques have proven an established ability to force flow reattachment over separated surfaces when actuation is applied continuously, the reattachment effects persist for the duration of actuation [1, 2]. When applied in short burst or as discrete singular input events, the impulse-like disturbance introduced by the flow control technique excites a broad spectrum of the separated flow which triggers multiple instabilities for an effect that persists substantially longer than the duration of actuation. However, a key feature shared among these impulse-like techniques is a transient lift response that bears a lift-reversal spike at the onset of actuation. Such a spike presents a challenge from the perspective of real-time flow control: this presents an inherent delay in the system, limiting the bandwidth of control [3].

The performance of a lifting body, certainly within an unsteady regime, is significantly altered by the dynamics of the near-body wake [4] be it induced by body motion, excited by application of flow control actuators, or naturally occurring feature of the flow. For a given static airfoil there exists a critical Reynolds number by which a Hopf bifurcation[5] occurs. The result is an oscillatory wake, in contrast to the steady, separated flow field of the sub-critical regime. At sufficiently high Reynolds number in the super-critical regime the oscillatory wake transitions to vortex shedding. When a given airfoil is driven in harmonic motion, synchronization between the rigid body's dynamics and the vortex shedding frequency may be exploited to yield departures from standard performance. Indeed, such resonant behavior was exploited by Choi *et al.* [6] in surge and plunge motions where time-averaged forces were significantly changed when the airfoil was driven at the vortex shedding frequency and its subharmonic. To similar effect, Dawson *et al.* [7] demonstrated substantial increases in aerodynamic loading in resonant pitching. Further, it was shown that substantial lift may be yielded for airfoil pitching at specific frequencies when the airfoil is separated but wake is steady. That is, there exist con-

*Email address: alberto.medina.3@us.af.mil

figurations in which vortex shedding being induced by airfoil motion can enhanced loading akin to resonant behavior. Such behavior was explore by Cleaver *et al.* [8] as a high-lift mechanism where low-amplitude plunge oscillations of an airfoil at zero angle of attack experienced stable deflected jets of vorticies responsible for large lift coefficients. The frequency of natural vortex shedding at elevated Reynolds numbers and angles of attack appears to satisfy a Strouhal range of $St = 0.13 - 0.20$, as demonstrated by Huang and Lin [9]. At such Reynolds numbers moderate attack angles are transitional and the wake does not reflect the Strouhal number criteria.

In this study, the response of a separated flow of a NACA 0006 wing to rapid-actuation of a conventional flap are explored. The airfoil is bisected about the midchord to a test article comprised of 50% leading edge and 50% trailing-edge flap. The leading-edge element is held fixed (α_{LE}) while the flap is allowed to pivot about the airfoil midchord (δ). The receptivity of varying degrees of separation to rapid step-and-hold maneuvers by the flap is examined by comparison of two representative flow states: $\alpha = 10°$ corresponding to a separated flow, and $\alpha = 20°$ corresponding to a massively-separated flow. In these comparisons, the flap executes a rapid step of $\pm 1°$ from an initial planar configures ($\alpha = \delta$). "Rapid" motions are performed within a fraction of a single convective time. We elect to employ mechanical actuation to avoid delay or "dead time" associated with conventional fluidic actuators, but instead seek to attain an instantaneous aerodynamic response to flap actuation, rooted in bound circulation following Wagner's solution to impulsive change in flap deflection [10]. Lastly, the effects of sustained low-amplitude harmonic flapping on the near-body wake shedding dynamics is explored for the separated case of $\alpha = 10°$ where we seek to impose authority over the shedding frequency for a flow regime that is historically transitional.

2 Experimental Setup

Experiments were performed in the Air Force Research Laboratory (AFRL) Horizontal Free-Surface Water Tunnel (HFWT). The tunnel is outfitted with a three-degree-of-freedom motion stage, consisting of a suite of linear motors, allowing the test article to engage in pitch, plunge, and surge maneuvers (or combination thereof). The test article is a NACA 0006 airfoil measuring 200mm in chord that spans the width of the water-tunnel test section (457.2mm) to produce a nominally two-dimensional flow field. The airfoil is bisected about the midchord, as shown in Fig. 1, to construct a trailing-edge flap of 50% chord. The resulting fore element is held fixed at a static incidence angle (α_{LE}) while the flap is pivoted about the midchord to execute a specified kinematic schedule.

Direct force measurements were acquired via two six-component force/moment balances. Both the fore element and the flap element were supplied a dedicated force balance. This arrangement allowed for the independent measurement of the constitutive elements' aerodynamic loading history. The fore element and the flap do not communicate mechanically, but instead are hydrodynamically coupled with a midchord gap of 0.5 mm separating the two. Planar flow visualization was conducted by the illumination of a fluorescent dye. Illumination was provided by laser sheet optics positioned at the three-quarter span position, and dye was introduced into the flow field through dye ports located at the leading and trailing edges of wing. Further detail can be found in Ref. [11, 12, 13].

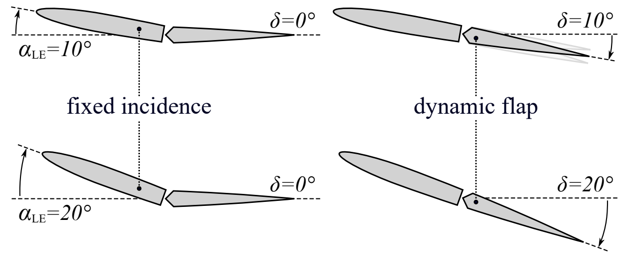


Figure 1: NACA 0006 test article configuration: (*top row*) $\alpha_{\mathrm{LE}} = 10°$, (*bottom row*) $\alpha_{\mathrm{LE}} = 20°$.

This study examines the response of separated flow to a excitation input in the form of rapid trailing-edge flap deflection at a chord-based Reynolds number of $Re = 4 \times 10^4$. Conveniently, the freestream speed and chord length provide for a convective time that equates to 1.0 s of wall-clock time. The distinction of "rapid" is reserved here for maneuvers by the flap that are completed in a fraction of one convective time. This study concerns discrete excitation of a singular step-and-hold maneuver by the flap as well as the effects of sustained actuation of the flap in harmonic deflection. For harmonic deflection the flap is driven in a sinusoidal schedule with a deflection amplitude of 1° (for $\alpha = 10°$: $\delta = 11 \to 9°$, for $\alpha = 20°$: $\delta = 21 \to 19°$) and frequency 6Hz ($TU/c = 0.167$). The step-and-hold maneuvers follows from a C^∞-smoothing ramp function formulated by Eldredge *et al.* [14]. Here, however, the smooth ramp has been fitted to a semi-period of a sinusoidal waveform, providing a step in deflection angle of $\pm 1°$. For $\alpha = 10°$ this entails a flap motion of either $\delta = 10 \to 11°$ or $10 \to 9°$. And for $\alpha = 20°$ this entails a flap motion of either $\delta = 20 \to 21°$ or $20 \to 19°$. The step in deflection angle by the smooth ramp, although small in amplitude, appears rather steep and is interpreted akin to an impulse in bound circulation. Because the geometry of the airfoil is tied directly to the mode of excitation, the deflection amplitude is selected such that deflection does not result in a geometry that deviates greatly from the baseline undeflected airfoil flow.

3 Results

3.1 Baseline Flow

We begin with examination of the baseline flow field states of interest, as depicted in Fig. 2. Two flow states are considered in assessing the potential control authority of the flap engaged in rapid maneuvers. These states correspond to $\alpha_{\rm LE} = 10°$ and $\alpha_{\rm LE} = 20°$ and represent prototypical separated and massively separated flow fields, respectively. In the instances of Fig. 2 the wing is held static in a freestream and the flap deflection angle (as measured with respect to the horizontal plane) is equal to the leading-element incidence angle ($\delta = \alpha_{\rm LE}$) to provide a planar airfoil. Both orientations $\alpha_{\rm LE} = 10°$ and 20° are distinguished by their separated shear layer emanating form the wing leading edge, where the case of $\alpha = 20°$ exudes further expulsion of the shear layer and expansion of the separation envelope. As anticipated, the time-averaged lift-to-drag ratio incurs a decrease with increasing severity of flow separation. At $\alpha_{\rm LE} = 10°$ aerodynamic performance measures $\overline{L/D} = 3.662$ and measures a reduced value of $\overline{L/D} = 2.547$ for 20°.

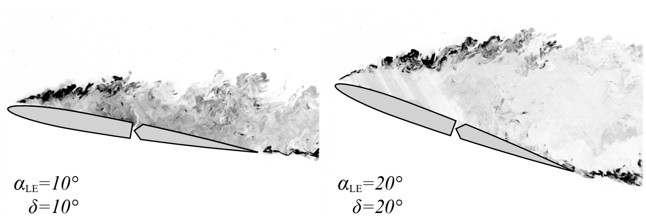


Figure 2: Baseline flow fields of static airfoils: (*left*) $\alpha_{\rm LE} = 10°$ separated flow, (*right*) $\alpha_{\rm LE} = 20°$ massively separated flow.

3.2 Transient Flap-Step Response

The effect of a rapid step in deflection angle by the trailing-edge flap is studied here through direct force measurement. As previously noted, the flap performs a step in deflection angle amounting to a pitch-and-hold maneuver. Because this maneuver is interpreted here as a surrogate for a bound-circulation impulse, the direction of flap pitch should induce directional-dependent responses as the circulation sign correlates with pitch direction. Thus, a small amplitude step is supplied of $\Delta\delta = \pm 1°$ from the initial planar configuration of $\alpha_{\rm LE} = \delta$.

Beginning with the separated case of $\alpha = 10°$, Fig. 3 shows the time history of lift in response to a flap rapid step. Lift is displayed as a differential value measuring the temporal deviation in lift from the initial static mean value. Mean values for all initial and final geometric configurations considered in this study are summarized in Table 1 for convenience. Prior to motion both the $+1°$ and $-1°$ lift curves track well with the time-averaged static lift value, indicated by the grey horizontal line. Upon executing the deflection step at $t^* = 5$ (where $t^* = tU/c$) there exists a rather sharp, yet brief, inertial spike. Directly thereafter the lift curve demonstrates an immediate response with a resulting lift value differing from the initial static value. The transient lift profile initiates with an effective vertical shift in value from the pre-deflection state. This immediate response is in contrast to the performance of some conventional fluidic actuators which when pulsed suffer from lag or deadband prior to realizing the desired response.

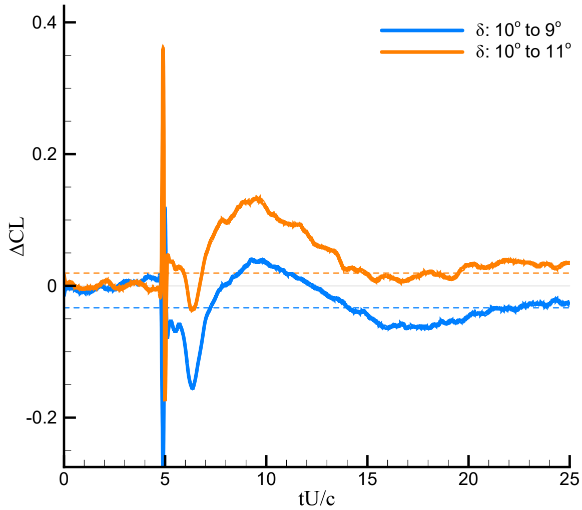


Figure 3: Transient response for $\alpha_{\rm LE} = 10°$: (*grey line*) $\overline{\rm CL}|_{\delta=10°} = 0.6943$. (*dash line*) Time-averaged static lift of final step angle.

$\alpha_{\rm LE}$	δ	$\overline{CL}$	$\overline{CL1}$	$\overline{CL2}$
10°	10°	0.6943	1.1629	0.2258
	9°	0.6619	1.1343	0.1859
	11°	0.7127	1.1601	0.2625
20°	20°	0.7954	1.1538	0.4399
	19°	0.7624	1.1148	0.4109
	21°	0.8556	1.2236	0.4852

Table 1: Time-averaged static lift values.

As previously discussed, the current mechanical mode of excitation is coupled with the geometry of the airfoil. Thus the long-time behavior of the lift curve in "hold" is best characterized as a relaxation to a new steady asymptotic value denoted in Fig. 3 as the dashed lines. In a $\Delta\delta = +1°$ maneuver, the terminal geometry provides for a net positive camber. Conversely, for $-1°$ there results a net negative camber. As such, the relaxation values of lift differ from the undeflected airfoil, with $\Delta\delta = +1°$ exceeding initial lift production ($\delta = 10°$) and $-1°$ generating a reduction in steady lift.

Despite differences in asymptotic relaxation lift values, the two modes of flap deflection exhibit significant transient responses with temporal histories bearing a remarkable resemblance. The transient lift profile trends among the two deflection cases initiate with a nominal plateau value which persists for approximately one convective time followed by a steep dip. Recovery from the dip is sharp, with the entire recovery event occurring over approximately one convective time. The lift profile continues to climb over the next three convective times to achieve a global transient peak, followed by a gradual relaxation over the next several convective times. Post inertial-spike, the relative magnitudinal differences from peak-to-peak lift appears preserved among the two cases. However, the two cases are distinguished by their initial vertical shift in lift. For the positive-camber motion, the transient lift profile is elevated to instantaneously exceed even the final static lift value of $\delta = 11°$. This is in contrast to the negative-camber motion where the transient profile appears to experience a downward shift from the pre-motion value, falling below even the static lift value of $\delta = 9°$.

As previously mentioned, the test article is equipped with a dedicated load cell per wing element. This configuration allows for inspection of the respective contributions to lift between the leading-edge element and the flap. This is particularly utile when only a subset of components of the wing's articulated body are driven in motion while the remaining components are held fixed in space. The fixed leading element's response comes as an artifact of its proximity to the driven flap through hydrodynamic coupling. The resulting transient lift histories are shown in Fig. 4 where CL1 refers to the leading element and CL2 the flap element. Lift coefficients here are normalized by their respective chordwise segment lengths (for this geometry $\hat{c} = c/2$). As previously observed, the transient profiles of both cases appear quite similar, with the resulting changes in lift differing nominally by way of vertical offset from the pre-motion value of lift. The flap suffers from a momentary anti-lift spike prompting a negative lift differential, ΔCL2, independent of the deflection direction. However, the vertical lift offsets (post inertial spike) remain directional-dependent. The apparent drop in CL (Fig. 3) for $\Delta\delta = -1°$ appears predominantly incurred by the fixed leading element where the apparent vertical offset in lift is most severe.

Thus far the influence of flap step-direction has been limited to the vertical offset in lift, bearing negligible command of the transient profile or response time for a separated flow corresponding to $\alpha = 10°$. Only when the flow field is progressed to a massively separated state does the step direction offer divergence in transient profile characteristics for the seminal cases of $\Delta\delta = \pm 1°$. To achieve the desired separation the airfoil is realigned with the leading element fixed at $\alpha_{\text{LE}} = \delta = 20°$ (Fig. 2). The lift response for the representative massively separated flow is show in Fig. 5. As before, prior to motion ($t^* < 5$) the phase-averaged lift tracks well with the time-averaged static lift value ($\overline{CL} = 0.795$). Barring the inertial spike at $t^* = 5$, the transient response to a flap step of $\Delta\delta = +1°$ is marked by the instantaneous realization of the long-time static lift value. It is without cunctation or developmental rise that the lift for a positive-camber step achieve the asymptotic value of the newly configured airfoil ($\delta = 21°$). This stands in contrast to conventional fluidic actuators which when employed in discrete pulsing induce an initial anti-lift spike upon actuation. Approximately one convective time after the motion is completed, the case of $+1°$ experiences successive peaks of diminishing magnitude with the initial lift peak occurring near $t^* = 7.5$.

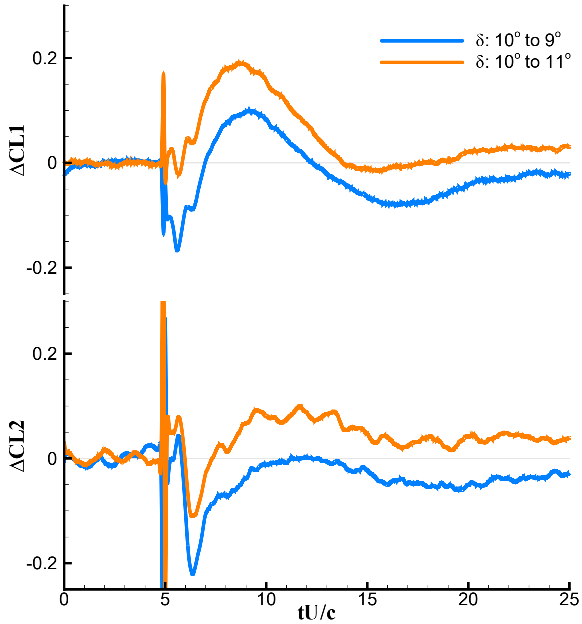


Figure 4: Transient response for $\alpha_{\text{LE}} = 10°$: (*top*) leading element CL1, (*bottom*) flap element CL2.

When the flap is driven in step toward a negative camber ($\Delta\delta = -1°$) the response garnered in lift is drastically altered from the positive-camber counterpart. The vertical negative shift in lift remains a prominent feature just as before in $\alpha = 10°$. The negative shift lies well below the static value of both $\delta = 20°$ and $19°$. However, approximately one convective time thereafter, the transient lift signature undergoes a substantial transformation in its aggressive upward surge. The steep rise in lift readily exceeds the initial undeflected static state ($\delta = 20°$), culminating in a global peak approaching $t^* = 7.5$. Note that the peak lift induced by the negative-camber step nearly doubles that of the positive-camber motion. During relaxation there exists a minor peak nearly coincident with the secondary peak cited for the positive-camber step approaching $t^* = 10$. In both step-direction cases the relaxation time remains intact. Relaxation is nominally achieved approximately 10 convective times after the deflection step is executed. This scaling is consistent with the separated case ($\alpha = 10°$) suggesting the transient dynamics scale with convective time.

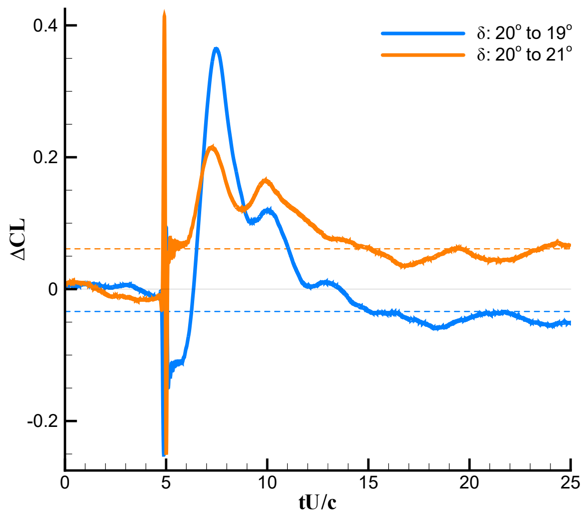


Figure 5: Transient response for $\alpha_{\mathrm{LE}} = 20°$: (*grey line*) $\mathrm{CL}|_{\delta=20°} = 0.7954$. (*dash line*) time-averaged static lift of final step angle.

The respective contributions to lift from both the leading- (ΔCL1) and flap element (ΔCL2) are displayed in Fig. 6. Inspection of the lift histories associated with $\Delta\delta = +1°$ and $-1°$ reveal striking commonalities between the two cases. Indeed, many of the overall trends among the two cases' profiles in Fig. 5 and Fig. 6 appear common, though with differing temporal scaling. It is understood that there exists some interaction between the flap and the separated shear layer for both the separated and massively separated flows. The mechanisms by which the step modes affect the flow fields governing the resulting transients is provided some clarity through flow visualization in the following section.

3.3 Flow Visualization of the Flap-Step Response

Investigation of the massively separated flow ($\alpha = 20°$) response to a negative-camber step ($\Delta\delta = -1°$) in Fig. 7 reveals large scale rollup of the leading-edge shear layer, as found in [12]. By $\Delta t^* = 0.56$ (where Δt^* is the duration of time elapsed since the completion of motion) the leading-edge free shear layer is severed from the separation envelope. The remains of the layer attached to the leading edge continues to rollup at $\Delta t^* = 0.72$, inducing flow toward the airfoil surface. In this manner, the flow becomes reattached with the formation of a leading-edge vortex. The shear layer continues to feed the leading-edge vortex at $\Delta t^* = 1.5$. During this period the remnants of the baseline separation envelope are convected along the chord, replaced by the advancing leading-edge vortex which continues to appreciate in size and extend the region of reattachment. By $\Delta t^* = 2.42$ the recirculatory region extends the entire length of the airfoil and

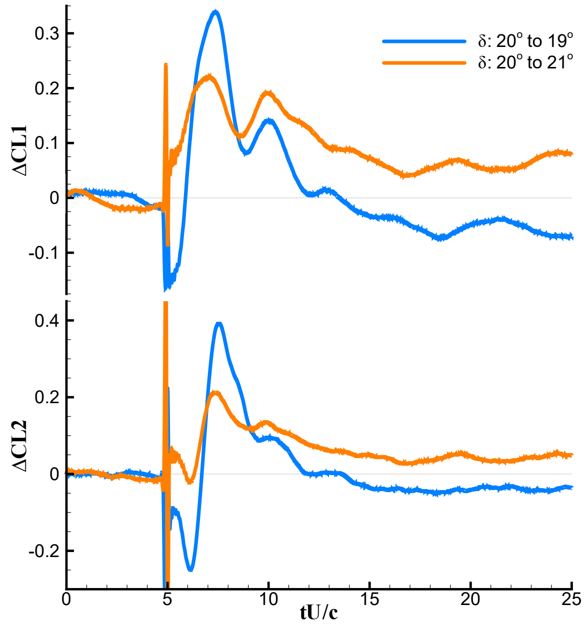


Figure 6: $\alpha_{\mathrm{LE}} = 20°$

coincides with peak lift production. Beyond this point the circulatory region is gradually removed from the airfoil surface, particularly the trailing-edge region, as the relaxation process converges to the massively separated state once again.

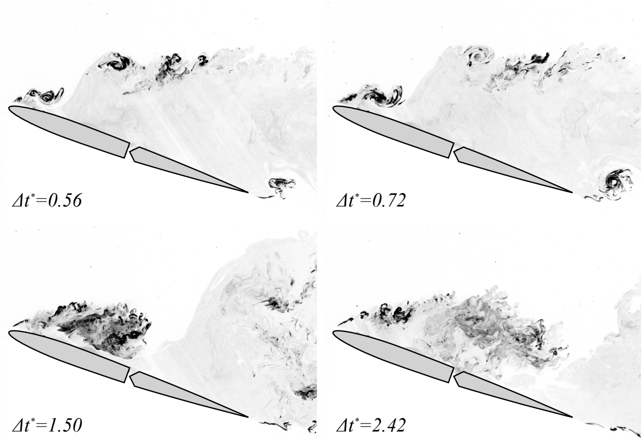


Figure 7: $\alpha_{\mathrm{LE}} = 20°$, $\delta = 20° \rightarrow 19°$, $6Hz$

Previously it was noted the transient lift profiles garnered in $\pm 1°$ step deflections differed primarily in temporal magnitude. As such, it would be rather plausible that the mechanisms by which the lift transients are produced in both step cases are quite similar and rudimental to massively separated flows. To this end, we note a comparable evolution of the flow for $\Delta\delta = +1°$ in Fig. 8. Shortly after the step at $\Delta t^* = 0.56$ there exists a distinct vortical formation within the shear layer near the leading edge. This formation, akin to that of the previous case, effectively disrupts the shear layer from feeding the separation envelope. As a result, a leading-edge vortex is created and proceeds to entrain the surrounding flow field toward the airfoil surface. By $\Delta t^* = 1.5$ the entrainment process does not appear as effective as the previous case of negative-camber step but provides a similar byproduct of shedding the baseline separation envelope.

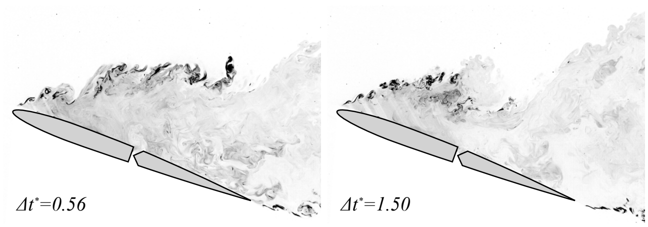


Figure 8: $\alpha_{\rm LE} = 20°$, $\delta = 20° \to 21°$, $6Hz$

The receptivity of the representative massively separated flow has become apparent, showcasing rollup of the free shear layer culminating in the formation of a leading-edge vortex independent of step direction. Returning focus back to the separated case of $\alpha = 10°$ the response of the flow is mitigated to flow reattachment. Fig. 9 provides flow field snapshots for a negative-camber step input. Upon completing the flap motion, the shear layer is drawn to the airfoil and ceases to feed the separation envelope. As time progresses the packet of vorticity drawn from the shear layer convects downstream, adhering to the contour of the airfoil surface. That is, the step of the flap is sufficient to momentarily reattach the flow for several convective times. As shown in Fig. 9, at $\Delta t^* = 2.42$ the entirety of the airfoil is now reattached with a well-behaved trailing edge shear layer. Although the flow field is devoid of large scale leading-edge vortex formations, the process of reattachment introduces significant structures as the initial separation envelope is supplanted by the reattached boundary layer. Upon relaxation the flow is returned to its baseline separated state.

To similar effect, a positive-camber step also proves disruptive to communication between the free shear layer and the separation envelope, as shown in Fig. 10. The ensuing boundary layer, however, appears to host larger vortical elements conducive to elevated lift production. Through these investigations it appears excitation by a low-amplitude rapid ramp of the trailing-edge flap deflection is sufficient to garner an immediate aerodynamic response. Within the representative separated flow field, only the positive-camber step of $\Delta\delta = 1°$ produced a net increase in transient lift. The response of the representative massively separated flow provided a tradespace between accelerated (instantaneous) realization of long-time lift and peak lift production offered by a positive-camber and negative-camber step, respectively.

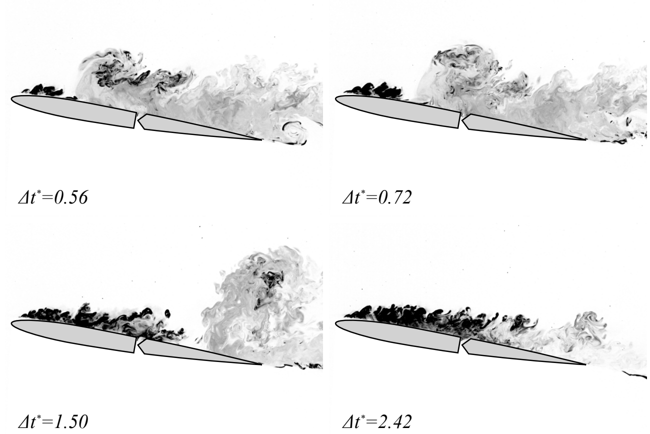


Figure 9: $\alpha_{\rm LE} = 10°$, $\delta = 10° \to 9°$, $6Hz$

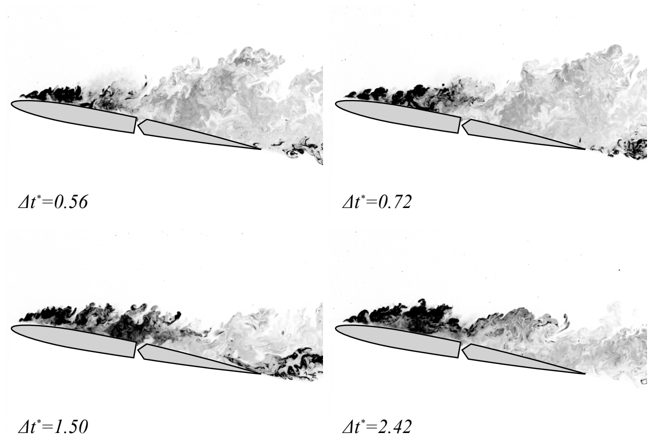


Figure 10: $\alpha_{\rm LE} = 10°$, $\delta = 10° \to 11°$, $6Hz$

3.4 Periodic Flap Actuation

Conventional fluidic actuators have showcased the utility of continuous or periodic excitation to great effect on separated flows. These demonstrations often entail establishing and maintaining reattachment under sustained excitation for steady cruise conditions at elevated angles of attack. As we have employed mechanical actuation of the flap as a surrogate for bound circulation pulsing the prospect of periodic actuation presents a dilemma: due to geometric constraints the flap cannot provide continuous excitation of desired direction. If sequential pulses/steps of $\Delta\delta = -1°$ are desired, for example, the flap would require a return phase of motion as the flap is reset in preparation for the subsequent pulse. With this consideration, this study proceeds with focus given to the separated state of $\alpha_{\rm LE} = 10°$ subjected to a sinusoidal flap schedule given the insensitivity of the transient profile (Fig. 3) to step direction and the relative agreement of transient flow structures among both step cases (Fig. 9, 10). The sinusoidal deflection cycle is performed about a mean flap angle of $\delta = 10°$ with a deflection amplitude of $\Delta\delta = 1°$, with a peak-to-peak sweep angle of $2°$ ($\delta = 11° \rightarrow 9°$). The frequency of flapping is derived from the desire to invoke a shedding event on the order of one convective time to affirm some measure of authority over the wake. This equates to a Strouhal number of $St = 0.175$ which is well within the reported bounds of shedding ($St = 0.13 - 0.20$) [9]. Such a display is made all the more remarkable considering the transitional state of the $\alpha = 10°$ near-body wake.

The phase-averaged lift history for a sinusoidal flap cycle is presented in Fig. 11. The phase-averaged lift curves were generated from a series of 100 flap cycles where the initial and final 10 cycles of measurement were discarded from the averaging scheme to remove starting transients and stopping effects. Among the actuation rates there is a general trend of increasing lift amplitude with increasing Strouhal number. Shortly after peak acceleration phases of motion ($t/T = 0.0$ and 0.5) trends show a global minimum and maximum. However, global lift peaks are not entirely in phase with acceleration and the overall lift profile exhibits quite a bit of undulation throughout the flap cycle indicating the forces generated are not superficially owed to dominance of non-circulatory added mass effects. Of these cases particular interest is given to the lift history of Strouhal number $St = 0.175$, corresponding to a frequency of $f = 1.0Hz$. That is, one flap cycle is completed over one convective time. Further, vortex shedding of the supercritical Hopf bifurcation regime is typically observed within $St = 0.13 - 0.2$. However, for $Re = 4 \times 10^4$ the near wake is understood to be in transition for an airfoil fixed at $\alpha = 10°$. The question remains as to whether the low-amplitude sinusoidal kinematics of the flap are sufficiently influential to impose some regularity or authority over the shedding pattern in the near-body wake to a representative case of $St = 0.175$.

A sense of repeatability of the lift cycle is provided by observation of Fig. 12 which presents the cycle-to-cycle variation of lift coefficient with deflection angle. Transient cycles (the first 10 cycles) are highlighted to differentiate from cycles included in the phase-averaging of Fig. 11. The black loops are considered fully-developed and relatively free of starting transients. Although transient cycles demonstrate a gradual drift toward the ‘fully developed’ periods there does not appear to be major transformations in hysteresis characteristics among the entirety of the run cycles. Indeed, once fully developed there does appear to be cycle-to-cycle variation in mean lift, however, importantly there does not appear to be discernible shift in phase between flap angle and lift history.

It is well understood that the near-body wake dynamics can bear a significant influence on aerodynamic loading. To examine the extent of lock-in, the near-body flow field of the wing is quantified by particle image velocimetry. The interrogation window is situated 2.5 chord downstream of the airfoil leading edge. A snapshot of the vorticity field is provided in the sub-diagram of Fig. 13. Initial inspection of vorticity shows a trail of vorticity packets convecting downstream. To quantify the the frequency content of this wake dynamic mode decomposition is performed, where the normalized mode amplitudes are presented in Fig. 13 along with the first four dynamic modes. Mode 1 reflects the signature of a mean vorticity field and is of the largest amplitude among the modes. Mode 2 is the second largest amplitude and corresponds to patterns congruent to the sizing of the vorticity concentrations. It is noted that the frequency associated with Mode 2 also matches the driving frequency of the sinusoidal flap maneuvers. This observation, coupled with the relatively abrupt decline of subsequent mode amplitudes, would indicate that by actuation of the trailing-edge flap some measure of authority may be retained, or rather reinstated, over the near-body shedding frequency even in a transitional operation space.

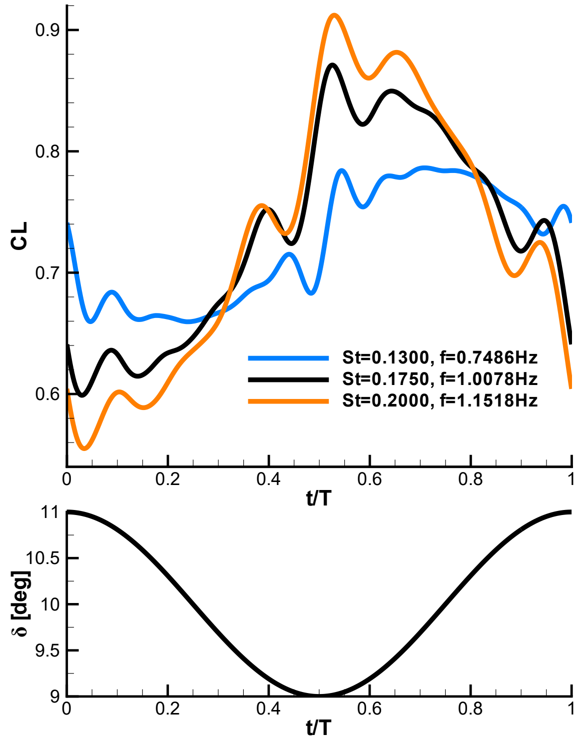


Figure 11: (*top*) Phase-averaged lift cycle variation with Strouhal number. (*bottom*) Flap deflection cycle.

4 Conclusion

The current effort to examine the response of a separated flow to rapid mechanical actuation of a trailing-edge flap included step-and-hold and continuous harmonic flapping maneuvers. In both kinematic maneuvers the amplitude of motion was limited to $1°$. In step-and-hold the flap was employed as a surrogate for a bound circulation impulse. It was found upon completing the maneuver the transient lift

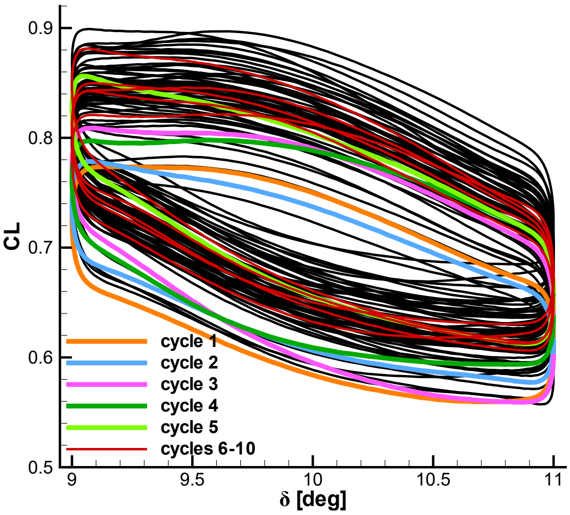


Figure 12: Lift hysteresis and cycle-to-cycle variation for $\alpha = 10°$ under sinusoidal flapping.

profile would experience an instantaneous shift in magnitude dependent on the direction of step. The resulting lift history for the representative separated case of $\alpha = 10°$ did not appear to display modification to the overall lift profile trends in response to flap step direction. However, for a massively-separated case of $\alpha = 20°$ the transient lift profile experienced temporal modification to scaling. It was also noted that a positive-camber step in both separated and massively separated states amounted to an instantaneous attainment of the long-time steady lift value associated with the final hold geometry. This is interpreted as a potential avenue to expand the limited bandwidth of conventional control techniques in the perspective of feedback control. The temporal scaling of profile magnitude experienced for the massively separated case appeared to correlate with the effectiveness of the flap to induce rollup of the leading-edge shear layer. When the flap was engaged in continuous harmonic pitching at $\alpha = 10°$ we were able to demonstrate authority over the near-body wake shedding behavior. The combination of attack angle and Reynolds number ensured the airfoil wake is in transition, external to the shedding guidelines of $St = 0.13 - 0.20$. However, quantification of the wake revealed the flap was sufficient to impose a shedding pattern corresponding to $St = 0.175$. Lock-in of the wake and large scale shedding was verified by application of dynamic mode decomposition.

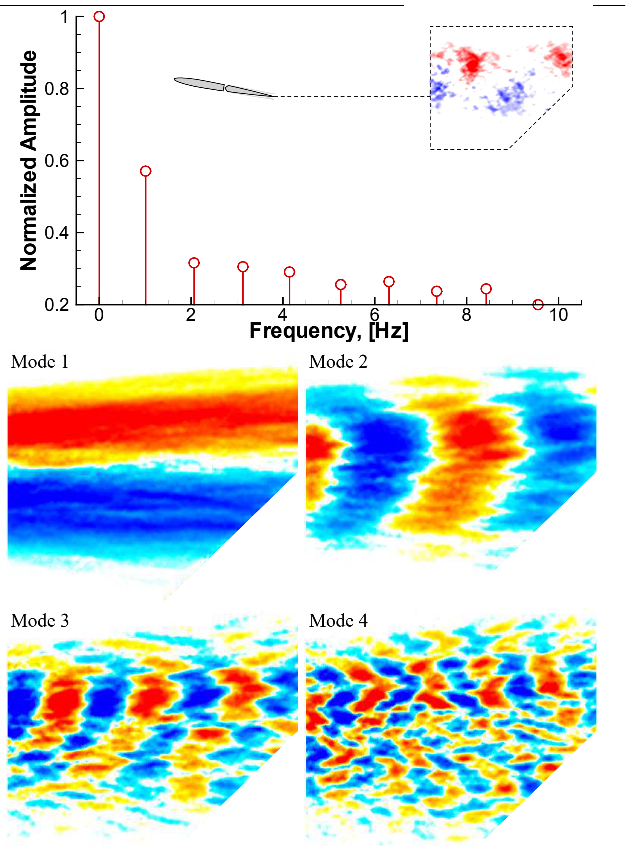


Figure 13: DMD of the near-body vorticity field: (*top*) normalized mode amplitude, (*bottom*) select modes.

References

[1] A. Seifert, T. Bachar, D. Koss, M. Shepshelovic, and I. Wygnanski. Oscillatory blowing: A tool to delay boundary layer separation. *AIAA Journal*, 31(11):2052–2060, 1993.

[2] A. Seifert, A. Darabi, and I. Wygnanski. Delay of airfoil stall by periodic excitation. *Journal of Aircraft*, 33(4):691–698, 1996.

[3] W. Kerstens, J. Pfeiffer, D. Williams, R. King, and T. Colonius. Closed-loop control of lift for longitudinal gust suppression at low reynolds numbers. *AIAA Journal*, 49(8):1721–1728, 2011.

[4] W. McCroskey. Unsteady airfoils. *Annual Review of Fluid Mechanics*, 14(1):285–311, 1982.

[5] K.R. Sreenivasan, P.J. Strykowoski, and D. J. Olinger. Hopf bifurcation, landau equation, and vortex shedding behind circular cylinders. In *Forum on Unsteady Flow Separation*, volume 52, pages 1–13, 1987.

[6] J. Choi, T. Colonius, and D.R. Williams. Surging and plunging oscillations of an airfoil at low reynolds number. *J. Fluid Mech.*, 763:237–253, 2015.

[7] S.T.M. Dawson, D.C. Floryan, C.W. Rowley, and M.S. Hemati. Lift enhancement of high angle of attack airfoils using periodic pitching. In *2016 AIAA Aerospace Sciences Meeting, AIAA SciTech Forum*, 2016.

[8] D.J. Cleaver, Z. Wang, and I. Gursul. Investigation of high-lift mechanisms for a flat-plate airfoil undergoing small-amplitude plunging oscillations. *AIAA Journal*, 51(4):968–980, 2013.

[9] R.F. Huang and C.L. Lin. Vortex shedding and shear-layer instability of wing at low-reynolds numbers. *AIAA Journal*, 33(8):1398–1403, 1995.

[10] H. Wagner. über die entstehung des dynamischen auftriebes von tragflügeln. *Zeitschrift für Angewandte Mathematic und Mechanik*, 5(1):17–35, 1925.

[11] A. Medina, M.V. Ol, P. Mancini, and A. Jones. Revisiting conventional flaps at high deflection rate. *AIAA Journal*, 55(8):2676–2685, 2017.

[12] A. Medina and M. Hemati. Separated flow response to rapid flap deflection. In *2018 AIAA Aerospace Sciences Meeting, AIAA SciTech Forum*, 2018.

[13] A. Medina, M.V. Ol, D.R. Williams, X. An, and M. Hemati. Modeling of conventional flaps at high deflection-rate. In *2017 AIAA Aerospace Sciences Meeting, AIAA SciTech Forum*, 2017.

[14] M .Ol, J. Eldredge, and C. Wang. High-amplitude pitch of a flat plate: an abstraction of perching and flapping. *International Journal of MAVs*, 1(3):203–216, 2009.

Safe corridor based task interface for quadrotors

Lai Shupeng*, Lan Menglu, and Ben M. Chen
National University of Singapore, 21 Lower Kent Ridge Rd, Singapore

ABSTRACT

Most of the task management interface of quadrotors is based on waypoints. While it is natural for the human to visualize the tasks and easy for quadrotors to execute, there lacks support for functions such as the geo-fencing. In many applications, it is desired to limit the quadrotors' operation area in the safe region. In this paper, we propose an approach that guides the vehicle in a predefined safe corridor without solving the entire trajectory beforehand. It also guarantees the feasibility by limiting the trajectory's velocity, acceleration, and jerk to a predefined range. The effectiveness of the proposed approach is demonstrated with real flight experiment.

1 INTRODUCTION

Quadrotors are used more and more frequently in industrial applications such as inspection, monitoring, and surveillance due to its agility and easy-to-maintain mechanical structure. The quadrotor's mission is usually described by a series of waypoints where it is expected to travel in sequence. Though the waypoint based missions are easy for the human to visualize and edit, it is a non-trivial task to ensure the resulting trajectory is still suitable. As in these applications, the quadrotor is usually required to be operated in a safe area with no apparent obstacles such as tall buildings, and its possible crash will have limited damage. Methods in [1] and [2] achieved this by building a safe-flying corridor connecting the waypoints. The size of the corridor can be adjusted, and the trajectory is restricted inside the corridor through constrained quadratic programming. This approach requires a more powerful on-board computer, especially in the case where a re-planning is needed, and the data link is not reliable enough thus planning on a remote computer is not an option. In this paper, we propose a safe corridor based task interface where the user could edit the mission quickly, and the safe trajectory could be generated efficiently which benefits the vehicles with weaker onboard computers. The rest of this paper is organized as the following. In Section 2, the safe-flying corridor used in our interface is introduced. In Section 3, we present an incremental approach to generate jerk, acceleration and velocity limited trajectory that stays inside the safe-flying corridor. And in Section 4, experimental results are analyzed and discussed. Finally, a conclusion is made in Section 5.

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

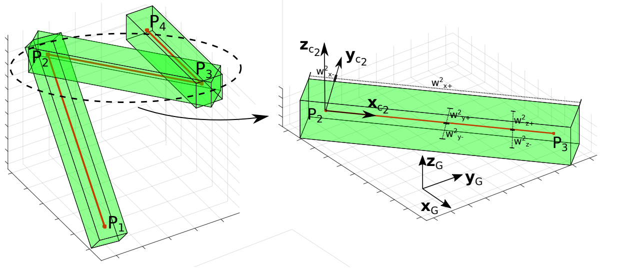


Figure 1: Nominal plan and flight corridors

2 SAFE-FLYING CORRIDOR

Given a list of waypoints, we call the line-segments constructed by connecting the waypoints in sequence as the nominal plan. And the safe-flying corridor is built around such a nominal plan. As shown in Figure 1, the nominal plan is defined by the waypoints P_1 to P_4 in the global frame $\mathcal{G}$. For each line segment defined by P_i and P_{i+1}, a local frame $\mathcal{C}_i$ is defined where its $x_{\mathcal{C}_i}$ axis is aligned to the vector $\overrightarrow{P_iP_{i+1}}$ and its $y_{\mathcal{C}_i}$ axis is perpendicular to the gravity direction. A safe bounding box (the green cuboid in Figure 1) aligned with the local frame is then adapted to enclose the line-segment. In this way, a safe region could be constructed around the nominal plan, and its size can be adjusted by setting the dimension of each safe bounding box. Given a trajectory $\mathbf{T}$, $t \in [t_0, t_f]$ in the 3 dimensional space as

$$\mathbf{T}(t) = \begin{cases} f_x(t) \\ f_y(t) \\ f_z(t) \end{cases}$$

we can check whether it is inside a safe bounding box by:

- Project the trajectory $\mathbf{T}$ into the local frame $\mathcal{C}_i$ of the safe bounding box as

$$\mathbf{T}_\mathcal{C}(t) = \begin{cases} f_{x_\mathcal{C}}(t) \\ f_{y_\mathcal{C}}(t) \\ f_{z_\mathcal{C}}(t) \end{cases}$$

- Find the minimum and maximum value of $f_{x_\mathcal{C}}, f_{y_\mathcal{C}}, f_{z_\mathcal{C}}$.
- Check whether all of the minimum and maximum values are inside the safe bounding box. If so, the entire trajectory $\mathbf{T}$ will be enclosed by the safe bounding box.

Using this method, we could decouple the enclosure checking problem into three extreme value searching problems. And if the trajectory is in the form of a polynomial, the searching could be done efficiently through root finding.

3 Incremental safe trajectory generation

3.1 *Jerk limited trajectory*

As a basic tool used in the incremental safe trajectory generation process, we present an approach based on the jerk limited trajectory appeared in [3], which is later improved in [4], and shown effective for quadrotors in [5]. In this paper, we present an approach that solves the position set-point problem using a direct bisection search, which does not require to build decision trees as in [3, 4]. And we also allow to set asymmetrical constraints on the velocity, acceleration and jerk.

3.1.1 Problem formulation

Given a triple integrator system

$$\begin{aligned} \dot{p} &= v \\ \dot{v} &= a \\ \dot{a} &= j \end{aligned} \tag{1}$$

where p, v, a, j are the position, velocity, acceleration and jerk respectively and the jerk j also serves as the system's input. The presented algorithm aims to bring the system in Equation 1 from an arbitrary initial state to a position set-point while satisfying constraints on the velocity, acceleration and jerk:

$$\begin{aligned} &p(0) = p_0, \quad p(t_f) = p_f \\ &v(0) = v_0, \quad v(t_f) = 0 \\ &a(0) = a_0, \quad a(t_f) = 0 \\ &v_{\min} \leqslant v(t) \leqslant v_{\max},\ \forall t \in [0, t_f] \\ &a_{\min} \leqslant a(t) \leqslant a_{\max},\ \forall t \in [0, t_f] \\ &j_{\min} \leqslant j(t) \leqslant j_{\max},\ \forall t \in [0, t_f] \end{aligned} \tag{2}$$

To make sure an solution does exist, it is assumed

$$\begin{aligned} v_{\min} &< 0 < v_{\max} \\ a_{\min} &< 0 < a_{\max} \\ j_{\min} &< 0 < j_{\max} \end{aligned}$$

3.1.2 Velocity set-point problem

Here, we first introduce the velocity set-point problem described in [3], the task is to bring the system in Equation 1 from an arbitrary initial state to n velocity set-point:

$$\begin{aligned} &v(0) = v_0, \quad v(t_f) = v_f \\ &a(0) = a_0, \quad a(t_f) = 0 \\ &a_{\min} \leqslant a(t) \leqslant a_{\max},\ \forall t \in [0, t_f] \\ &j_{\min} \leqslant j(t) \leqslant j_{\max},\ \forall t \in [0, t_f] \end{aligned} \tag{3}$$

Unlike in [3], in our formulation, we allow asymmetrical limits on the acceleration and jerk. Our solution is based on the one in [3], and it is shown in Algorithm 1. First, we try instantly bring the acceleration to zero, check whether the resulted v_e is larger or smaller than the desired velocity v_f and determine the cruise direction of the acceleration profile (line 3 – 8). Then depends on the cruise direction, we try to bring the acceleration either to its maximum or minimum value, and check whether the resulted velocity over or undershoots v_f (line 9 – 37). If it undershoots, there will be an cruise phase with non-negative time endurance (line 36), otherwise we solve for the switching acceleration (line 43 and 48) depending on the cruise sign. The final result is an parameter structure holding the desired jerks and their corresponding endurance. For simplicity, we use the function

$$\mathcal{P} = \text{solveVelocity}(v_0,\ a_0,\ a_{\max},\ a_{\min},\ j_{\max},\ j_{\min},\ v_f)$$

to denote the calculation of $\mathcal{P}$ in algorithm 1. With the initial state p_0, v_0, a_0 and $\mathcal{P}$, it is straight forward to reconstruct the trajectory through the model in Equation 1. The function

$$(p_s, v_s, a_s) = \text{getState}(v_0, a_0, p_0, \mathcal{P}, t_s)$$

is used to calculate the state of the trajectory (p_s, v_s, a_s) at a specific time-point t_s.

3.1.3 Position set-point problem

Now, we extend the solution to cover the position set-point problem in Equation 2 which has been studied in [3] and [6]. Our method is based on a bisection search to find the solution rather than using decision trees. The detail of our algorithm can be seen in Algorithm 2.

To identify the cruise direction, we first solve for the braking trajectory that immediately brings the velocity and acceleration both to zero (line 3). The resulted stopping point p_{sp} is used to determine the cruise velocity by comparing with the desired target p_f (line 3–12). Then we create the zero cruise profile by steering the system to the cruise velocity and immediately to full stop (line 13–16).

The resulting stop point might over or undershoot the p_f. If it undershoots the desired position, then the non-negative cruise time can be found as in line 19. And if it overshoots the desired position, then the cruise velocity cannot be reached, and we switch the system towards zero speed before reaching v_c. The exact switching time is found through a bi-section search (line 22–38). With the parameters for the two different phases $\mathcal{P}_a$, $\mathcal{P}_b$ and the switching time t_{pb}, the trajectory can be constructed using the triple integrator model. An example of such a trajectory with asymmetrical constraints on its velocity, acceleration and jerk is given in Figure 2.

Algorithm 1 Velocity target solver

Input: v_0, a_0 ,$a_{\max}$,$a_{\min}$,$j_{\max}$, $j_{\min}$, v_f
Output: $\mathcal{P}$
if $a0 \geq 0$ **then**
  $v_e = v_0 + a_0 \left| a_0/j_{\min} \right| /2$
else
  $v_e = v_0 + a_0 \left| a_0/j_{\max} \right| /2$
end if
$d_a = \text{sign}(v_f - v_e)$
if $d_a == 1$ **then**
  $a_c = a_{\max}$
else if $d_a == -1$ **then**
  $a_c = a_{\min}$
else
  $a_c = 0$
end if
if $a_c - a_0 \geq 0$ **then**
  $t_1 = (a_c - a_0)/j_{\max}$
  $j_1 = j_{\max}$
else
  $t_1 = (a_c - a_0)/j_{\min}$
  $j_1 = j_{\min}$
end if
$v_1 = v_0 + a_0 t_1 + t_1^2 j_1/2$
if $-a_c \geq 0$ **then**
  $t_3 = (-a_c)/j_{\max}$
  $j_3 = j_{\max}$
else
  $t_3 = (-a_c)/j_{\min}$
  $j_3 = j_{\min}$
end if
$\bar{v}_3 = a_c t_3 + t_3^2 j_3/2$
$\bar{v}_2 = v_f - v_1 - \bar{v}_3$
if $d_a == 0$ **then**
  $t_2 = 0$
else
  $t_2 = \bar{v}_2/a_c$
end if
if $t_2 < 0$ **then**
  if $d_a == 1$ **then**
    $a_n = \sqrt{(2(v_f - v_0) + a_0^2/j_{\max})/(1/j_{\max} - 1/j_{min})}$
    $t_1 = (a_n - a_0)/j_{\max}$
    $t_2 = 0$
    $t_3 = -a_n/j_{\min}$
  else if $d_a == -1$ **then**
    $a_n = -\sqrt{(2(v_f - v_0) + a_0^2/j_{\min})/(1/j_{\min} - 1/j_{max})}$
    $t_1 = (a_n - a_0)/j_{\min}$
    $t_2 = 0$
    $t_3 = -a_n/j_{\max}$
  end if
end if
$\mathcal{P}.T_1 = t_1$
$\mathcal{P}.T_2 = t_2 + t_1$
$\mathcal{P}.T_3 = t_3 + t_2 + t_1$
$\mathcal{P}.j_1 = j_1$
$\mathcal{P}.j_2 = 0$
$\mathcal{P}.j_3 = j_3$

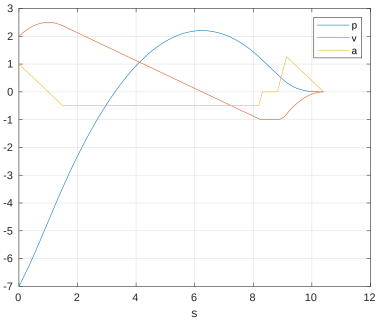


Figure 2: Trajectory with asymmetrical constraints. The position set-point is at zero position.

Algorithm 2 Position target solver

Input: $p_0, v_0, a_0, v_{\max}, a_{\max}, j_{\max}, v_{\min}, a_{\min}, j_{\min}, p_f$
Output: $\mathcal{P}_a, \mathcal{P}_b, t_{\text{pb}}, v_c, t_c$
$\mathcal{P} = \text{solveVelocity}(v_0, a_0, a_{\max}, a_{\min}, j_{\max}, j_{\min}, 0)$
$(p_{\text{sp}}, v_{\text{sp}}, a_{\text{sp}}) = \text{getState}(v_0, a_0, p_0, \mathcal{P}, \mathcal{P}.T_3)$algorithmic
$d_p = \text{sign}(p_f - p_{\text{sp}})$
if $d_p == 1$ **then**
  $v_c = v_{\max}$
else if $d_p == -1$ **then**
  $v_c = v_{\min}$
else
  $v_c = 0$
end if
$\mathcal{P}_a = \text{solveVelocity}(v_0, a_0, a_{\max}, a_{\min}, j_{\max}, j_{\min}, v_c)$
$(p_{\text{fa}}, v, a) = \text{getState}(v_0, a_0, p_0, \mathcal{P}_a, \mathcal{P}_a.T_3)$
$\mathcal{P}_b = \text{solveVelocity}(v_c, 0, a_{\max}, a_{\min}, j_{\max}, j_{\min}, 0)$
$(p_{\text{fb}}, v, a) = \text{getState}(v_c, 0, p_{\text{fa}}, \mathcal{P}_b, \mathcal{P}_b.T_3)$
$t_c = 0$
if $\text{sign}(p_{\text{fb}} - p_f) \cdot d_p \leq 0$ **then**
  $t_c = \dfrac{(p_f - p_{\text{fb}})}{v_c}$
  $t_{\text{pb}} = \mathcal{P}_a.T_3$
else
  $t_c = 0$
  $tH = \mathcal{P}_a.T_3$
  $tL = 0$
  for $counter = 1 : N$ **do**
    $t_{\text{pb}} = (tH + tL)/2$
    $(p_{\text{pb}}, v_{\text{pb}}, a_{\text{pb}}) = \text{getState}(v_0, a_0, p_0, \mathcal{P}_a, t_{\text{pb}})$
    $\mathcal{P}_b = \text{solveVelocity}(v_{\text{pb}}, a_{\text{pb}}, a_{\max}, a_{\min}, j_{\max}, j_{\min}, 0)$
    $(p_{\text{fb}}, v, a) = \text{getState}(v_{\text{pb}}, a_{\text{pb}}, p_{\text{pb}}, \mathcal{P}_b, \mathcal{P}_b.T_3)$
    if $\text{sign}(p_{\text{fb}} - p_f) \cdot d_p < 0$ **then**
      $tL = t_{\text{pb}}$
    else
      $tH = t_{\text{pb}}$
    end if
    if $|p_{\text{fb}} - p_f| < \epsilon$ **then**
      break
    end if
  end for
end if

To test the efficiency of the proposed method, the initial position, velocity and acceleration is set to be in the range of $[-50, 50]$, $[-10, 10]$ and $[-5, 5]$ with an incremental of 0.05. Without loss of the generality, the position set-point is always zero. And the $v_{\max}, a_{\max}, j_{\max}, v_{\min}, a_{\min}, j_{\min}$ is set as $4, 4, 2, -1, -1, -1$ accordingly. A total of 160880400 trajectories are generated and the average computing time is 0.31 microseconds with an i5-3470S CPU at 2.9 GHz.

3.2 Navigation in the safe corridor

Using the proposed jerk limited trajectory generation algorithm, an online and incremental approach is proposed to generate a safe trajectory that stays fully inside the safe corridor. As in Figure 1, let L_i denotes the line-segment defined by waypoints P_i and P_{i+1}, the corresponding local frame is $\mathcal{C}_i$ and the safe bonding box that enclose L_i is denoted B_i.

3.2.1 Generate trajectory in the local frame

In the local frame $\mathcal{C}_i$, we can generate a trajectory R_i that starts from an arbitrary state and stops at waypoint P_{i+1} by solving the position set-point problem on each axis of $\mathcal{C}_i$ (namely $x_{\mathcal{C}_i}$, $y_{\mathcal{C}_i}$ and $z_{\mathcal{C}_i}$) independently. Since the origin of $\mathcal{C}_i$ is at waypoint P_i and the target is P_{i+1}, the position set-point on the $x_{\mathcal{C}_i}$ axis is $\|P_{i+1} - P_i\|$. And the position set-points on $y_{\mathcal{C}_i}$, $z_{\mathcal{C}_i}$ are 0. Moreover, we have to assign the velocity, acceleration and jerk limits on each axis of $\mathcal{C}_i$, namely $x_{\mathcal{C}_i}, y_{\mathcal{C}_i}, z_{\mathcal{C}_i}$. As shown in [5], the physical limits of the quadrotor can be satisfied by limiting the trajectory's velocity, acceleration and jerk separately. However, these desired limits are usually defined in the global frame G and need to be projected into $\mathcal{C}_i$. Let $v_{\mathcal{G}} = [v_{\mathcal{G}_x}, v_{\mathcal{G}_y}, v_{\mathcal{G}_z}]$ denote the velocity in the global frame, a common practice is to have

$$\begin{aligned} \sqrt{v_{\mathcal{G}_x}^2 + v_{\mathcal{G}_y}^2} < v_{\mathrm{h}_{\max}} \\ v_{\mathrm{v}_{\min}} \leq v_{\mathcal{G}_z} \leq v_{\mathrm{v}_{\max}} \end{aligned} \tag{4}$$

because the quadrotor have similar dynamics in its horizontal axes compared to the vertical axis. The constraints span a cylindrical volume shown in Figure 3. However, in the $\mathcal{C}_i$ frame, the velocity constraints need to be decoupled into each individual axis. Let $v_{\mathcal{C}_i} = [v_{\mathcal{C}_{i_x}}, v_{\mathcal{C}_{i_y}}, v_{\mathcal{C}_{i_z}}]$ represent the velocity in the local frame, the constraint is

$$v_{\mathcal{C}_{i\lambda_{\min}}} \leq v_{\mathcal{C}_{i\lambda}} \leq v_{\mathcal{C}_{i\lambda_{\max}}}, \forall \lambda \in \{x, y, z\}$$

which spans an axis-aligned cuboid $Q_{v,i}$ in $\mathcal{C}_i$. Therefore, it is necessary to select the limits such that the cuboid is entirely inside the cylinder (see Figure 3). The same axis-decoupling criterion also applies to the acceleration and the jerk. Furthermore, the width of the spanned cuboid (Figure 3) needs to be large enough on all axis, so that the vehicle could move agilely towards any direction at any moment. It is crucial if an evasive maneuver is needed which might deviate from the planned trajectory.

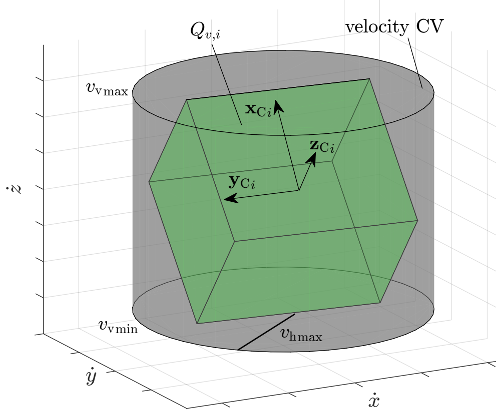


Figure 3: The volume spanned by constraints at $\mathcal{G}$ and $\mathcal{C}_i$.

3.2.2 Continuous navigation

With the capability to reach an arbitrary waypoint in the desired frame, we now consider navigating the quadrotor through multiple safe flying corridors. A trivial approach is to fly along each line-segments and stops at each waypoint. Here, we propose an approach to guide the vehicle inside the safe corridor without stopping at each waypoint. Assume there are total M waypoints and the vehicle is initially inside B_1, the proposed algorithm can be expressed in Algorithm 3. The idea is to repeatedly generate a new trajectory R_{i+1}

Algorithm 3 Smooth navigation

```
1: i = 0
2: while Not reaching P_M do
3:   if i < M then
4:     s = get_current_reference()
5:     s̄ = project(s, C_{i+1})
6:     R_{i+1} = generate(s̄, C_{i+1}, P_{i+1})
7:     if exam(R_{i+1}) then
8:       Start tracking R_{i+1}
9:       i = i + 1
10:    end if
11:  end if
12: end while
```

that connects the vehicle from its current reference state s to the new position set-point P_{i+1}, and then exam whether this new trajectory is safe. We first get the current reference state the vehicle is tracking (line 4), then it is projected into the local frame $\mathcal{C}_i$ and the new trajectory is generated from $\bar{s}$ to set-point P_{i+1} (line 5 – 6). And once R_{i+1} is considered safe, the vehicle could start to track it and proceed to try the next waypoint (line 7 – 10). While the implementation of get_current_reference() and project() is straight forward, the generate() function adopts the method in Section 3.2.1. Finally, the exam() function checks whether the tra-

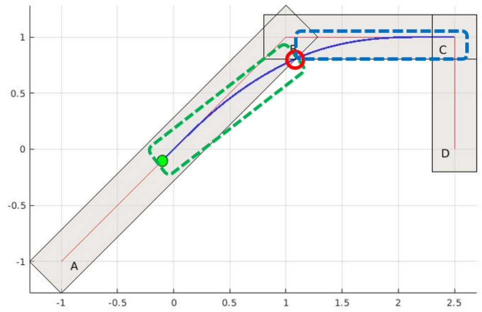


Figure 4: Split the trajectory into two segments, then check whether each individual part is inside a single bounding box.

jectory is fully inside the safe corridor and the satisfaction of constraints in the global frame $\mathcal{G}$.

3.2.3 Safety check

In [1], the author first samples the trajectory at multiple time instances, and then check each sample individually. Here, a continuous checking method is adopted due to the fact the trajectory consists of finite segments of third order polynomials. The process is illustrated in Figure 4 and can be summarized as the following:

1. Find split points that are contained within more than one bounding boxes (the red circle in Figure 4).
2. Split the trajectory into multiple segments (the dotted line rectangle in Figure 4).
3. Check whether at least one bounding box fully contains each of the split segment through finding the extreme values (see Segment 2).

The cylinder-shaped velocity, acceleration and jerk constraints (see Figure 3), can also be checked through finding the extreme values. Taking the velocity constraint from Equation 4 as an example, its satisfaction can be tested by:

1. Project the trajectory into the global frame $\mathcal{G}$.
2. Calculate the horizontal speed profile $v_\mathrm{h} = \sqrt{v_{\mathcal{G}_x}^2 + v_{\mathcal{G}_y}^2}$.
3. Find the extreme values of v_h and $v_{\mathcal{G}_z}$.
4. Check whether the extreme values is inside the constraint volume.

For a shorter checking time, it is also possible to only consider the trajectory that crosses at most 2 bounding boxes, thus limiting the amount of segments to be examined.

4 Experiments

To test the proposed algorithm, we perform an real experiment flight using an mini-quadrotor. The task (see Figure 5) involving reaching three targets (A, B and C) in sequence inside a pre-generated flying corridor among multiple obstacles. However, before the vehicle reaches target A, the task is modified to reach targets B and C only. The velocity, acceleration and jerk limited trajectory is generated online using the proposed method. And the video of the flight experiment can be found at https://www.youtube.com/watch?v=zAbCmOHj1EI. From Figure 5, our methods respond to the change of targets immediately and generate a trajectory that leads the vehicle to fly back towards target B. The trajectory is entirely inside the safe corridor throughout the process.

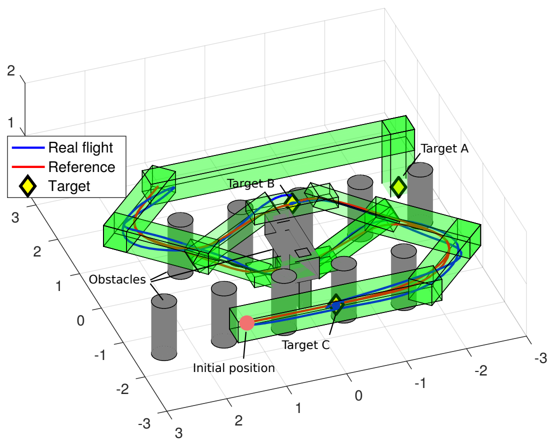


Figure 5: Experiment with multiple targets.

5 Conclusion

In this paper, we present an efficient algorithm to generate velocity, acceleration and jerk limited trajectory with asymmetrical constraints in detail. We then propose an approach to utilize the trajectory generation algorithm to guide a quadrotor to fly smoothly through a pre-generated safe flying corridor without non-necessary stopping. Our approach is efficient and could handle changes in the tasks with real-time responses. Compared to methods which require to solve a constrained quadratic optimization problem, our method is expected to be more suitable for vehicles with limited computational power.

References

[1] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, “Planning dynamically feasible trajectories for quadrotors using safe flight corridors

in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, July 2017.

[2] J. Chen, T. Liu, and S. Shen, "Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1476–1483.

[3] R. Haschke, E. Weitnauer, and H. Ritter, "On-line planning of time-optimal, jerk-limited trajectories," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2008, pp. 3248–3253.

[4] T. Krger, "Opening the door to new sensor-based robot applications; the reflexxes motion libraries," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1–4.

[5] M. Hehn, R. Ritz, and R. D'Andrea, "Performance benchmarking of quadrotor systems using time-optimal control," *Autonomous Robots*, vol. 33, no. 1, pp. 69–88, Aug 2012.

[6] T. Kröger and F. M. Wahl, "Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 94–111, Feb 2010.

Carbon fibre/PVC foam sandwich composite modelization for MAVs & long range drones structures

M. LEFEBVRE*, R. KIEFER, M. VEDRINES, and H. CHIBANE
Icube, INSA, CNRS, Strasbourg, France

ABSTRACT

This paper deals with the characterization, modelization and simulation of a composite structure used for MAV or generally for drones design. Composite structures are extremely difficult to simulate due to the anisotropic behaviour. The first part of the article is focused on carbon fiber/PVC foam (AIREX) sandwich composite characterization with the design of experiments method on tensile tests. This method gives equations, which describe the material mechanical behaviour (Young's modulus, tensile strength) depending on factors values. The second part of the article deals with the improvement of mechanical simulation of an anisotropic material with the goal to get an accurate model and to generalised properties to an entire structure. Then, the macroscopic mechanical properties of the most performing sandwich will be obtained with the global composite behaviour matrix 6x6, in order to be integrated into a Finite Element Analysis software (CREO / Simulate) to simulate a fixed wing behaviour. Finally, comparisons between experiment and numerical simulation on the wing will give promising results, and simulations will reveal substantial differences between tension & compression in a flexural solicitation.

1 INTRODUCTION

The design of experiments method (Or Taguchi method) is used to study various combinations of sandwich composite. Commonly used in aeronautic design, this method can be implemented in drones design because drones become more and more competitive. This method provides many advantages for drones' design, such as lean sizing and extra weight avoidance to increase drones' performances and reduce manufacturing cost [4].

This approach permits to investigate the effect of each factor on mechanical properties, and offers a way to choose the best sandwich material in order to follow the specifications. Design of experiments finally gives equations that take design factors into account. INSA STRASBOURG team CIGOGNE (https://www.facebook.com/equipecigogne/; https://fondation.unistra.fr/projet/imav2018/) uses this method in the design process of the ELCOD project INTERREG in cooperation with German and French universities. The objective is to design a low-cost long endurance drone. By choosing the correct carbon fibre sandwich for the structure and knowing the correct mechanical behaviour of the material, the objective is to choose the best factors combination that optimizes cost and high strength of the drone.

2 The design of experiments

The design of experiments used is a three factors general full factorial design.

- The first qualitative factor is carbon fibre type (Four levels: A, B, C and D)
- The second quantitative factor is fibre orientation (Two levels: 0° and 45°)

*contact email: martin.lefebvre@insa-strasbourg.fr

- The third quantitative factor is PVC foam (AIREX© sheet C70.75 – R&G composite) thickness (Three levels: 1.2mm, 2mm and 5mm)

Thus, for this experiments plan, 24 combinations were tested with tensile tests. Furthermore, for each combination, three experiments were made to test the repeatability of the measure and to have a convincing average response. Those experiments were also made on an article for the IMAV 2015 where the tensile norm and experiments process are clearly explained [1].

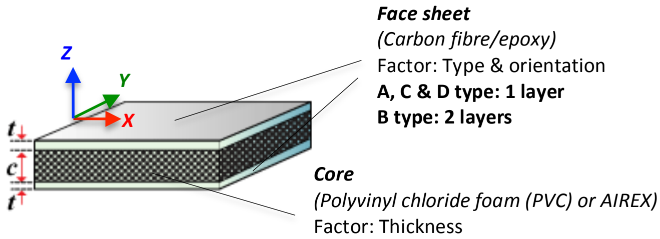


Figure 1 - Sandwich composite structure

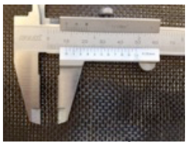
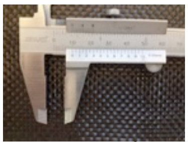
Carbon fibre type		
A	C	D
		
65 g/m²	**80 g/m²**	**160 g/m²**
Plain weave (60 tex)	Spread tow fabric TeXtrem (800 tex)	Plain weave (200 tex)

Table 1 - Carbon fibre type

- *The B type is made with two layers of carbon fibre 65g/m² (A type) for the face shee*
- *0° is the XX direction:*

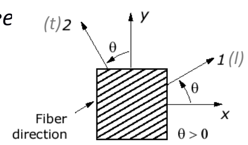


Figure 2 - Fibre orientation

F0	F1 (θ)	F2 (c)		
Carbon fibre type	**Orientation**	**Foam thickness [mm]**		
A	0°	1,2	2	5
	45°	1,2	2	5
B	0°	1,2	2	5
	45°	1,2	2	5
C	0°	1,2	2	5
	45°	1,2	2	5
D	0°	1,2	2	5
	45°	1,2	2	5

Table 2 - Design of experiments; factors' values

3 Results

3.1 Effect of factors

Tensile tests results are analysed to study each effect of factors on mechanical properties: Tensile strength and Young's modulus are the responses studied.

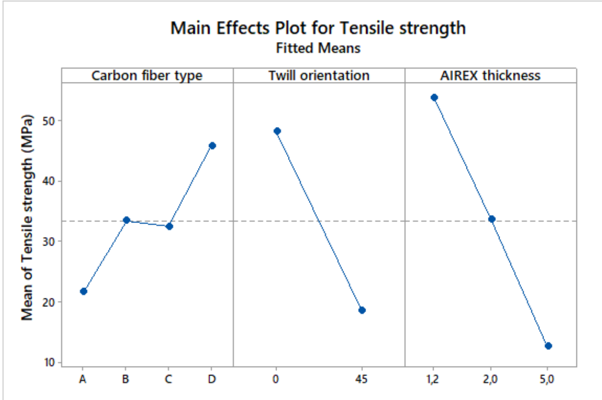


Figure 3 - Effect of each factors on average tensile strength

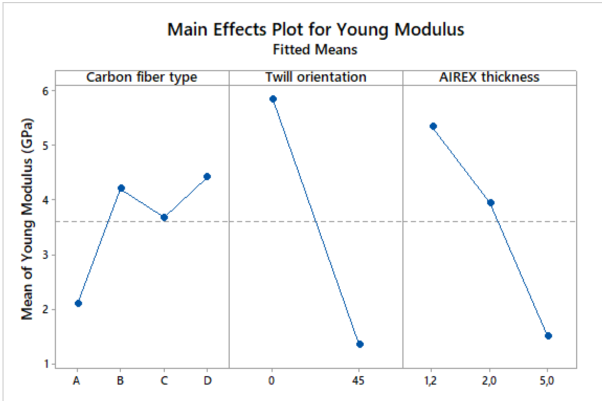


Figure 4 - Effect of each factors on average Young's modulus

When PVC foam thickness increases, average mechanical properties decrease, because sandwich properties converge to foam properties. That is adequately explained with the sandwich rule of mixture [3]. Off-axis orientation decreases also mechanical properties [2, 5].

With this method, a selection can be done:

- A type hasn't got enough mechanical strength
- B type is too expensive in a low-cost design (the carbon fibre surface is doubled compare to the other types)
- 80g/m² and 160g/m² carbon (C & D type) have to be preferably considered in terms of mechanical properties. In the following, a complete analysis will be done.

3.2 Composite behaviour equations

3D plots give visual results on the best sandwich material composition. The comparison between sandwich C and D is done on the Figures 5 & 6. Behaviour equations are given on the Equation 1 & 2.

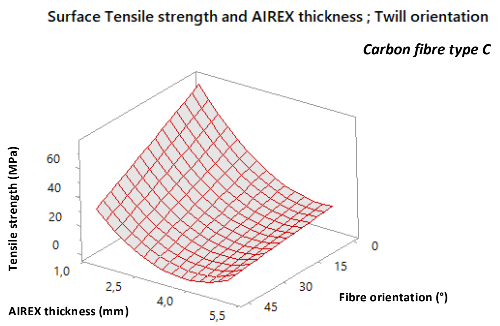


Figure 5 - Carbon fibre C type; factors 3D plot

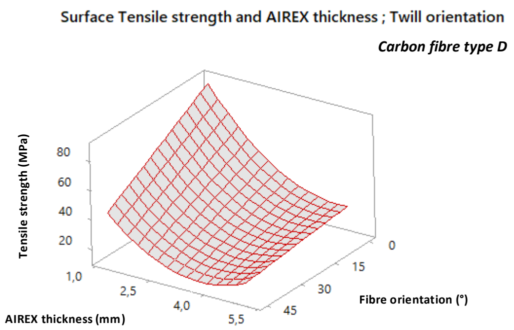


Figure 6 - Carbon fibre D type; factors 3D plot

Sandwich C
$\sigma_{max} = 122{,}1 - 1{,}23{\times}F1 - 45{,}0{\times}F2$ $+4{,}76{\times}F2^2 + 0{,}21{\times}F1F{\times}2\ (eq.1)$
Sandwich D
$\sigma_{max} = 135{,}5 - 1{,}23{\times}F1 - 45{,}0{\times}F2$ $+4{,}76{\times}F2^2 + 0{,}21{\times}F1{\times}F2\ (eq.2)$

Table 3 - Composite behaviour equations (R^2=81,04%)
F1: Orientation (θ) - F2: Foam thickness (c)

Equation 1 & 2 gives theoretical tensile strength for this factors combination. It is an important time saving in the design step.

Example: For the carbon fibre C, with θ=0° and foam thickness=3mm:

$$\sigma_{max} = 122{,}1 - 1{,}23{\times}0 - 45{\times}3 + 4{,}76{\times}3^2 + 0{,}21{\times}0{\times}3 = \mathbf{40{,}94}\,\boldsymbol{MPa}$$

Design experiments method can be use for structure sizing. By choosing factors values for the composite, it is possible to evaluate the theoretical Young's modulus and tensile strength.

4 Flexural models

The objective is to determine the most adequate flexural model to use in the following study. A 3-points bending test will be done to study flexural behaviour. As sample, a 2mm thickness sandwich beam C type will be tested.

4.1 Castigliano model

Equations 3 & 5 give the deflection Δ [mm] of a composite sandwich beam for a 3-point bending test under a load F [N]:

- $\Delta = \frac{Fl^3}{48(EI)_{éq}} + \frac{Fl}{4}\frac{k}{(GS)}$ (eq. 3) [2]

With $(EI)_{eq} = \frac{E_f bt^3}{6} + \frac{E_c bc^3}{12} + \frac{E_f btd^2}{2}$ (eq. 4) [6]

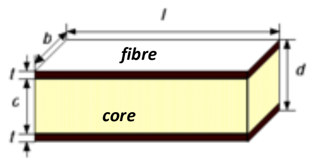


Figure 7 - Sandwich composite thickness data

This model takes the separate layers and the shear contribution (second term of Equation 3) into account.

For the bending test, properties of the sample are:

- $E_f = 49$ and $E_c = 0{,}066$ [GPa]
- $b = 50$, $l = 600$, $c = 2$ and $t = 0{,}22$ [mm]
- $k = 1$

Note: E_f and E_c are determined with suppliers' data and rule of mixture for E_f with epoxy resin. The error is quantified to +/-5%.

4.2 Homogeneous beam model (Voigt model)

- $\Delta = \frac{Fl^3}{48(EI)_{global}}$ (eq. 5)

The term $(EI)_{global}$ is determined with the tensile test from the design of experiments.

4.3 Experimental 3-points bending test

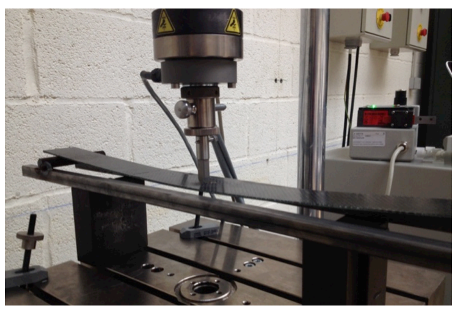


Figure 8 - 3-point bending test on sandwich sample (C type)

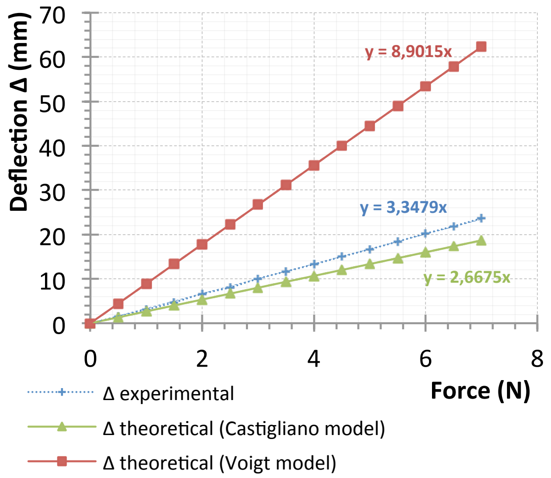


Figure 9 - Beam comparison between flexural models

Model	Relative error
Castigliano	20%
Voigt	166%

Table 4 - Relative error comparison between flexural models

In flexural solicitation, it's more adequate to use the Castigliano model (Sandwich model): To take the distinct layers into account with the $(EI)_{éq}$ term. The theoretical result will be closer to the reality with this model. For tensile, the macroscopic model can be used; it is by definition the behaviour of a composite with the rule of mixture. However, for flexural solicitation a more elaborate bending theory is needed [2, 7]. This accurate theory takes behaviour of heterogenic layers and the shear contribution into account. That is why the relative error is less important.

5 Best factor combination

Low foam thickness means high tensile strength and high Young's modulus. And high foam thickness means high rigidity (high second moment of inertia). The objective is to find a compromise between high tensile strength and high rigidity.

Thanks to the experiments, the optimization process gives a thickness between **1.8mm and 2,5mm** (white area on the Figure 10) with the carbon fibre C and 0° orientation to insure best compromise between high bending stiffness and high tensile strength:

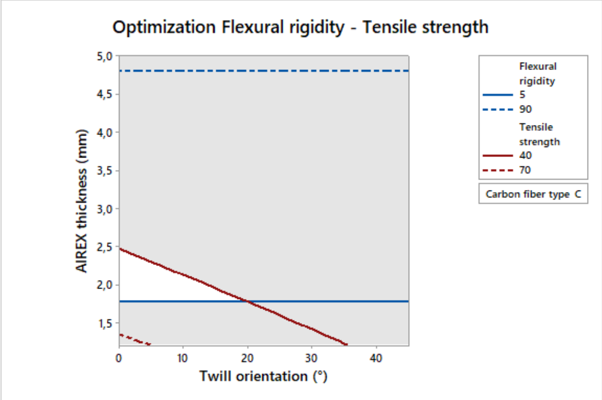


Figure 10 - Composite thickness optimization

In the following parts, a comparison will be done between a real bending test and a numerical simulation on a C type carbon fibre sandwich wing (NACA MH 32 wing profile) with a carbon fibre sandwich spar. The objective is to study the accuracy of a numerical simulation on complex fixed wing geometry. Thanks to the previous results, the choice of a 2mm foam, Carbon fibre C type is done for the analysis. The multi-layers model will be used to increase the accuracy of the simulation.

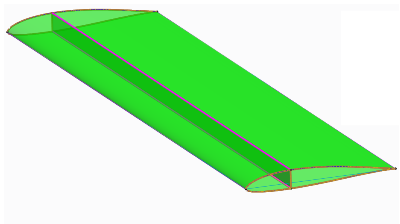


Figure 11 - Experimental wing MH32 + spar Composite sandwich structure C

6 Sandwich mechanical behaviour

The following equations are used for input all material data on CREO Simulate.

$$\begin{cases} E_x(\theta) = \dfrac{1}{\frac{c^4}{E_l}+\frac{s^4}{E_t}+c^2s^2\left(\frac{1}{G_{lt}}-2\frac{\nu_{tl}}{E_t}\right)} \\ E_y(\theta) = \dfrac{1}{\frac{s^4}{E_l}+\frac{c^4}{E_t}+c^2s^2\left(\frac{1}{G_{lt}}-2\frac{\nu_{tl}}{E_t}\right)} \\ G_{xy}(\theta) = \dfrac{1}{4c^2s^2\left(\frac{1}{E_l}+\frac{1}{E_t}+2\frac{\nu_{tl}}{E_t}\right)+\frac{(c^2-s^2)^2}{G_{lt}}} \end{cases} \quad \text{(eq. 6)} \quad [2]$$

Equations 6 can give G_{lt} thanks to the tests at $\theta = 45°$. Thus, all coefficients of stiffness matrix $[Q]$ can be determined for each layer:

$$[Q]_{Facesheet} = \begin{bmatrix} 58{,}47 & 14{,}03 & 0 \\ 14{,}03 & 58{,}47 & 0 \\ 0 & 0 & 30{,}47 \end{bmatrix}_{0°/90°}$$

$$[Q]_{Foam} = \begin{bmatrix} 0{,}073 & 0{,}022 & 0 \\ 0{,}022 & 0{,}073 & 0 \\ 0 & 0 & 0{,}028 \end{bmatrix}$$

(Units in GPa)

Distinct layers have to be integrated into the behaviour matrix 6x6, in order to get macroscopic sandwich composite behaviour:

$$[N] = \begin{bmatrix} A_{11} & A_{12} & 0 & | & B_{11} & B_{12} & 0 \\ A_{12} & A_{22} & 0 & | & B_{12} & B_{22} & 0 \\ 0 & 0 & A_{66} & & 0 & 0 & B_{66} \\ & — & & & & — & \\ B_{11} & B_{12} & 0 & & D_{11} & D_{12} & 0 \\ B_{12} & B_{22} & 0 & | & D_{12} & D_{22} & 0 \\ 0 & 0 & B_{66} & & 0 & 0 & D_{66} \end{bmatrix} \quad \text{(eq. 7)}$$

With:

$$A_{ij} = \sum_{k=1}^{n} Q_{ij}{}^{k} e_k \quad (8)$$

$$B_{ij} = \frac{1}{2}\sum_{k=1}^{n} Q_{ij}{}^{k}\left(h_k{}^2 - h_{k-1}{}^2\right) \quad (9)$$

$$D_{ij} = \frac{1}{3}\sum_{k=1}^{n} Q_{ij}{}^{k}\left(h_k{}^3 - h_{k-1}{}^3\right) \quad (10)$$

Note: For a symmetric composite: $B = 0$

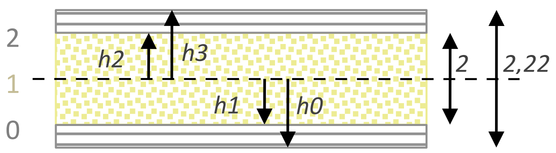


Figure 12 - Sandwich composite layers thickness

Finally, the carbon fibre/foam 2mm sandwich composite macroscopic behaviour can be expressed with this matrix:

$$[N] = \begin{bmatrix} 13009 & 3131 & 0 & & & & \\ 3131 & 13009 & 0 & | & & 0 & \\ 0 & 0 & 6759 & & & & \\ & — & & & & — & \\ & & & & 14379 & 3453 & 0 \\ & 0 & & | & 3453 & 14379 & 0 \\ & & & & 0 & 0 & 7486 \end{bmatrix}$$

(A matrix in N/mm; D matrix in N.mm)

7 Sandwich simulations

After computing matrix behaviour data (Equation 7) into CREO Simulate, flow stress and beam is calculated with a 10N load at the edge of the tested wing:

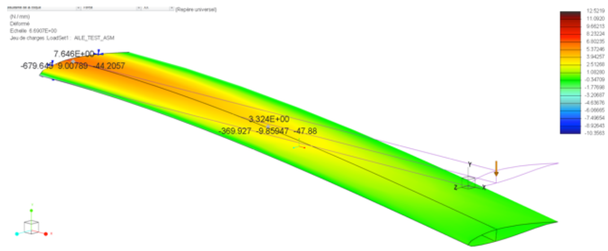


Figure 13 – Flow stress XX (N/mm) simulation on sandwich composite C; wing + spar (Display beam scale: 10%)

The upper side of the wing is in tension and the lower side in compression. The maximum stress is reach in the spar of the wing. It is this part on the wing which takes up efforts.

8 Sandwich experimental test

The objective is to compare simulation with an experimental bending test:

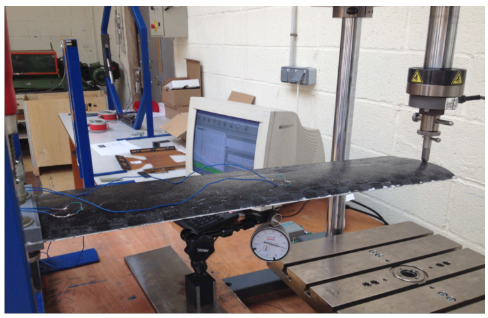


Figure 14 - Experimental Zwick Roell bending test bench

Strain is measured in three points with strain gauges:

Pt1: Upper side, close to rigid link

Pt2: Lower side, close to rigid link

Pt3: Upper side, middle of the wing

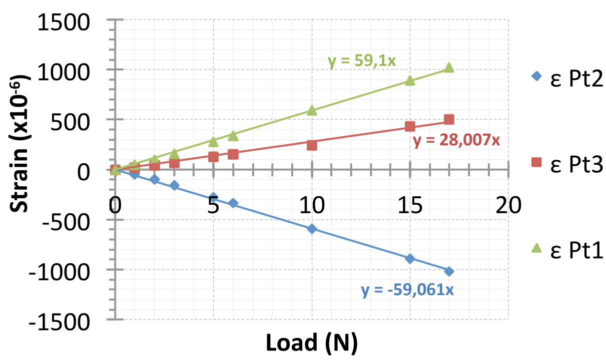


Figure 15 - Experimental wing strain on 3 points

In the experimental test, the tensile stress on the upper side of the wing is the same than the compressive stress on the lower side of the wing.

For compression the relative error is more important because it is difficult to predict compressive behaviour for fibre due to crushing, buckling and fibre delamination. Those results are showed in the Table 5.

	Numerical	Experimental	
	Stress XX (MPa)		Error
Pt1	3,83	3,95	3,1%
Pt2	-3,15	-3,94	25%
Pt3	1,66	1,63	1,8%
	Beam UY (mm)		Error
Max	10,31	13,62	25%

Table 5 - Relative error on stress and beam comparison

9 CONCLUSION

Design of experiments has made possible the choice the best sandwich composite for the corresponding application and gives equations to correctly design the composite structure. This method has to be used to design drones; it is an important time saving to compare structures and to choose the best factors' values.

After choosing the adequate sandwich composite, numerical simulations show that composite behaviour is not the same in tension or compression for high deformations. Even if the model with distinct layers is more accurate than the homogeneous model, investigation can be done to improve this theory. Buckling and fibre delamination can occur for compression, and a more exhaustive theory that takes layers slippage and plastic deformation into account has to be used. The flexural behaviour is non-linear for high compressive deformations [7]; and for the experiment of the present article, increasing the load to produce higher deformations could permit to study a more accurate non-linear compressive behaviour and reveal buckling [7, 8]. To go further with simulation, Tsai-Wu failure criterion can be used to study the limit of the composite, plus observation on sample have to be done to investigate damage and failure [8]. For numerical simulation, it is possible to be confident with tension, because the behaviour is linear-brittle and error is under 5%. However, it is highly recommended to be careful when local compression appears. Indeed, the current model is neither accurate nor optimal and simulation error is higher than 25% because of buckling.

REFERENCES

[1] Renaud K., Marc V., Proposition of design method and optimization of multirotors composite structures, ICube UMR 7357, 2015.

[2] D. Gay, Matériaux composite *(Composite Materials)*. Paris: Lavoisier Hermes, 6th revised edition, 2015. 687 p. (ISBN 978-2-7462-4707-9)

[3] CES EduPack, Sandwich Panel: Assumption and calculations, Granta, 85 p. 2017.

[4] R. Rikards, A. Chate, Optimal design of sandwich and laminated composite plates based on the planning of experiments, Structural Optimization, Riga Technical University, 1995.

[5] Kawai M., Morishit M., Satoh H., Tomuras S., Effect of end-tape shape on strain field of unidirectional carbon/epoxy composite specimens subject to off-axis tension, Composite Part A, Vol. 28A, pp 267-275, 1997.

[6] K. Joon Yoon, C.K. Kim and Hoon C. Park, Nonlinear flexural deflection of thermoplastic foam core sandwich beam, Journal of Composite Material, Vol. 36, No 13/2002, 2001.

[7] I. M. Daniel, J. L. Abot, K. A. Wang, Testing and analysis of composite sandwich beams, ICCM, 1999.

[8] E.E. Gdoutos, I.M. Daniel, Failure modes of composite sandwich beams, Thoeret. Appl. Mech., Vol.35, No.1-3, pp. 105-118, Belgrade 2008.

ACKNOWLEDGEMENTS

The ELCOD project (www.elcod.eu) is co-funded by the European Regional Development Fund (ERDF) and the co-financed project partners Region Grand Est and the countries of Baden-Württemberg and Rhineland-Palatinate in the framework of the INTERREG V Upper Rhine program.

Topological Optimization applied towards the development of a small and lightweight MAV composite frame

T.R.Costa[1,2] *, M. P. Caldas[1,2] †, P. M. N. Araujo[1,2] ‡and E. C. Silva[1,3] §
[1]AeroRio, Pontifical Catholic University of Rio de Janeiro, PUC-Rio, Rio de Janeiro, Brazil
[2]Department of Mechanical Engineering, Pontifical Catholic University of Rio de Janeiro, PUC-Rio, Rio de Janeiro, Brazil
[3] Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro, PUC-Rio, Rio de Janeiro, Brazil

ABSTRACT

This paper proposes the design and manufacturing process of a lightweight MAV composite frame with high structural efficiency, applying the topology optimization by minimizing the structure compliance. The study presents a real MAV frame, designed to the 2018 IMAV indoor competition. Structural optimization is a frequently used discipline in aerospace applications, and the topology optimization is its the most recent branch, that is used to obtain the optimal material distribution in a predefined domain. The utilized formulation tries to maximize the stiffness of the MAV composite frame, with stress constraints, in order to achieve high payload-to-empty-weight ratio, and energetic efficiency improving the vehicle autonomy.

1 INTRODUCTION

Several international market researches are forecasting a huge grown of the drone market in the next decade, since drones are demonstrating to be extremely useful to many agricultural, commercial, military and industrial applications. The study of unmanned vehicle systems is fundamental to ensure optimal results and reduce human risks, looking for greater efficiency, control and maneuverability.

In the context of the development of UAVs, there is a gradual growth of the research areas related to the mechanical-structural development of a multi-rotor drone frame. In that way, using a topological optimization method on the center plate of a MAV is of main importance, as structural and energetic efficiency are main goals for the development of an indoor competition for MAVs. This method is widely used in aeronautics, and the objectives sought are similar: the balance between stiffness and structural weight. Thus, the context of the structural efficiency, it is fundamental to go through a process of iterations related to three main aspects: the selection of geometry via topological optimization, the selection of materials by finite elements method and the study of the processes related to the main structures of the drone.

The development and application of the methods explained here are centered on the participation of the AeroRio UAV Design team in the International competition of International Micro Air Vehicles (IMAV) 2018. This competition aims to develop autonomous drones capable of performing a series of tasks involving both intelligence and the *de facto* structure of the drone. Thus, due to the restrictions involving the tasks and the scores of the competition, the maximum dimensions of the frame were estimated, which would allow the execution of the tasks. The developed MAVs need to be optimized for structural efficiency taking into account static and modal analysis for optimized structure validation.

The flight score of the IMAV 2018 indoor competition is directly affected by two multiplier factors, the Mass factor (M) and Power factor (W). The Mass factor increases as the MAV mass decreases, and the power factor increases by reducing the power capacity of the batteries, challenging teams to seek for structural and energy efficiency. In order to allow the developed MAV to navigate autonomously by the indoor course, proper embedded electronics (sensors, cameras, flight controllers, computer modules, etc) should be selected, aiming at weight and power consumption minimization. Besides, the propulsion system (motors and propellers) should be as efficient and lightweight as possible.

Regarding energy efficiency, must be said that by having a defined motor-propeller system, it is necessary that the frame has a high structural efficiency for the least power consumption of the motors for the displacement of the drone. It should be noted that by increasing the overall weight of the frame structure, the capacity of the batteries should be increased as well, since the motors will be operating under more extreme conditions. Thus, in developing a light and rigid structure, it is possible to obtain lower power consumption, as well as improvements in control and dynamics of the drone, considering the defined motor-propeller system.

The design of the frame contributes to the development of a small, lightweight and low power MAV. However, reducing the weight of the frame might lead to a reduction of its stiff-

*Email address: thiagorodric@gmail.com
†Email address: marcospereiracaldas@gmail.com
‡Email address: pablomilheiro@gmail.com
§Email address: edusilva@ele.puc-rio.br

ness, since there is a trade-off between stiffness and weight. The development of the drone frame follows the composite material selection and configuration, considering materials such as carbon fiber, plywood and Polylactic Acid (PLA). The maximum dimensions of the frame were estimated based on restrictions involving the tasks and the scores of the competition, aiming to ensure that it would be capable to execute all course elements.

In this paper, the adopted formulation for topology optimization seeks stiffness maximization by minimizing the structure compliance, for a given amount of material, in order to obtain a small and lightweight composite frame with high structural efficiency, providing improved control and stability. The developed MAV was optimized for structural efficiency, taking into account static and modal analysis for structural validation. The manufacturing procedure of the optimal structure will be also presented and discussed.

2 Configuration of the MAV

The indoor competition of IMAV 2018 highlights three main tasks which the drone must perform during the course: crossing windows of defined dimensions, crossing a maze of cylindrical obstacles and crossing hoops of restricted dimensions as well. Dimensional constraints are mainly generated by the smallest hoop present in the IMAV circuit, where it measures around 36 x 40 cm, so the MAV should have dimensions smaller than those. In addition, the MAV must be compact enough to accommodate the electronic and transmission system for carrying out the mentioned tasks. In this way, the MAV is restricted to lateral dimensions of less than 33 cm, as a safety factor. Thus, certain parameters must be chosen for the good performance of the drone. Among them, the configuration of the motor-propulsion system. Thus, after a series of tests the E-MAX MT1806 motor was chosen, since its mechanical dimensions were appropriate, besides allowing good possible thrust for the drone. The selection of the propeller was also restricted by the smaller hoop size, in which two were the most feasible to use, the 5” of diameter and the 6 ”. Thus, static thrust tests were performed to measure the effective thrust of the drone with the chosen motor-propulsion system. Thus, a maximum thrust of 270 g was obtained with the E-MAX MT1806 motor and the 5 ” three blade propeller.

The MAV must be capable of carrying a battery that provides enough power to the motors and sufficient current for the electronic image processing and control system. Thus, taking into account the choice of the propulsion system, a 2S of 5200 mAh was chosen, allowing around 300 g of thrust. Thus, it should be noted that stiffness involves both the manufacture of a frame that is bending moments resistant and having the first high frequency modes, away from the approximately 150 Hz generated by the motors. The bending strength is fundamental to allow higher efficiency of the motors, as these will lose less power for the deformation of the frame.

3 Optimization

The Topology Optimization Method have been widely employed for sizing and shape optimization of aerospace structures [1], since it can adapt the structural configuration for its restrictions by redistributing the material layout and accordingly the load carrying paths. This technique has been developed since the Bendsøe and Kikuchi [2], specially for least-weight and performance design.

For this project, in order to achieve high payload-to-empty-weight ratio, a 2D frame with fixed thickness was optimized utilizing the Topology optimization tool in ANSYS Mechanical, in order to minimize the structure compliance. The compliance basic formulation is presented on the following equation (1) based on [3], which maximize the stiffness of the MAV frame.

$$\begin{cases} \min\limits_{x}: & c(x) = U^T K U = \sum_{e=1}^{N} (x_e)^p u_e^T k_0 u_e \\ \text{subject to}: & \frac{V(x)}{V_0} = f \\ : & KU = F \\ : & 0 < x_{min} \leq x \leq 1 \end{cases} \tag{1}$$

From the equations we have that, **F** and **U** are respectively the global force and displacement vectors, **K** is the global stiffness matrix, u_e and k_e are the element displacement vector and stiffness matrix, respectively, **x** is the vector of design variables, $x_m in$ is a vector of minimum relative densities (non-zero to avoid singularity). Also, **p** is the penalization power, $V(x)$ and V_0 is the material volume and design domain volume, respectively and **f** is the prescribed volume fraction.

The initial domain dimensions of the frame and the position of the motors were defined respectively by the smallest hoop present in the IMAV circuit, and the propeller size required for the desired thrust. In order to optimize the computational effort, forces and moments were applied to half frame due to symmetry, as shown in Figure 1 where the free body diagram is presented.

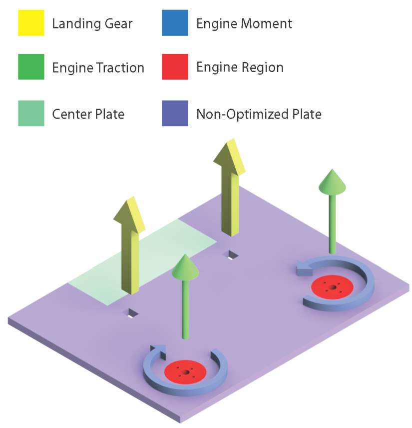


Figure 1: Free body diagram for optimization

In order to obtain the optimal structure, the Static Structural analysis were realized with the frame fixed support being the center plate, subjected to several steps and combinations of loads, mainly due to the motor and the landing gear of the drone as presented in Table 1. magnitude of the loads were determined by the selected propulsion configuration of the Drone.

Structure	**Type**	**Loads**
MOtor	Max Thrust	3.0 N
	Max Moment	0.5 N.m
Landing Gear	Landing impact	4.5 N

Table 1: Main Loads for Static Structural ANSYS analysis

The regions excluded from the optimization domain are in red, and were defined by the Boundary Conditions of the Static Structural analysis, as can be seen on Figure 2

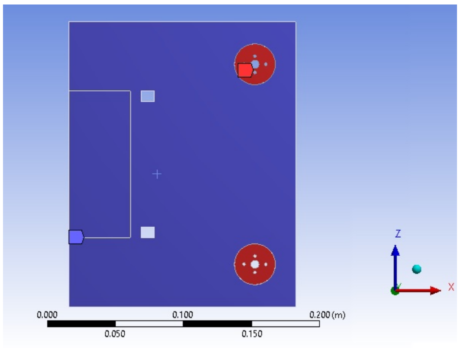


Figure 2: Optimization Region

The Topology Optimization was realized with the parameters tabulated on the following Table 2. The result obtained from the optimization is on Figure 3 and was used as model for the design of the frame that will be validated on the next section.

Element Type	Hexahedrons
Number of Elements	18450
Element Order	Quadratic
Max Number of Iterations	500
Convergence Accuracy	0.1%
Penalty Factor	3
Objective	Minimize Compliance
Response Constraint	Volume
Percent to Retain	30%
Member Min. Size	0.015 m

Table 2: Topology Optimization parameters

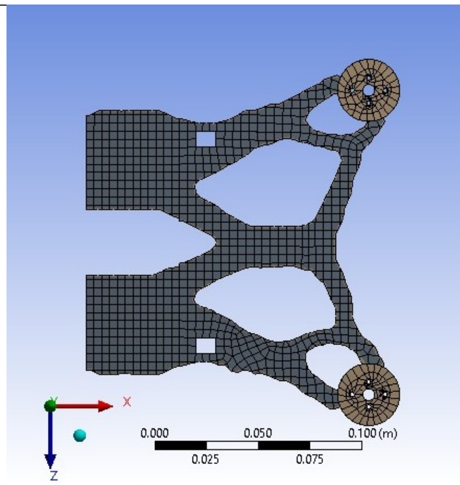


Figure 3: Topology Optimization result

4 Material Selection

Following the topological optimization, the materials were chosen considering the mechanical properties and the manufacturing processes to develop the frame. Several materials were considered, but some were highlighted because of the wide knowledge of their manufacturing techniques and applications in the aeronautical engineering. The materials, therefore, must follow the maximum design constraint: increased structural efficiency. Thus, materials highlighted and analyzed should be light and easy to manufacture, allowing to obtain geometries with smaller tolerances and to validate results of the optimization. In addition, they should confer high strength-to-specific-mass ratios. A good choice is composite materials, which allow the combination of diverse mechanical properties and the possibility of conformation to obtain the optimized geometry.

Thus, by choosing composite materials, there is the need to make two effective choices: the core and the reinforcement materials. The reinforcement is the component of the composite material that will suffer the major loads of the structure in question. The reinforcement must, therefore, possess such mechanical properties as high tensile and compressive strength, ie, mechanical properties related to the stresses suffered by the structure, in this case, the MAV frame. In that way the reinforcement must have a high mechanical resistance, combined with a low density, therefore following the maximum of a high specific resistance to the traction, fundamental in the case analyzed here.

Table 3 presents the mechanical properties of materials widely used in aeronautical industry that can be applied in structures such as the frame.

Carbon fiber has a high specific tensile strength giving important properties such as rigidity and resistance to loads. It must be considered that the model of the frame to be constructed must possess a fundamental characteristic that is the manufacturability. Carbon fiber has lamination methods that

Property	Unit	CFRP	PLA
Tensile Strength	MPa	600	46.8
Compressive Strength	MPa	570	46.8
Young Modulus	GPa	70	600
In-Plane Shear Strength	MPa	90	-
In-Plane Shear Modulus	MPa	5000	3350
Density	kg/m^3	1600	1290

Table 3: Mechanical Properties of the analyzed structural materials

allow the fabrication of complex geometry structures, but with certain constraints. Thus, the PLA presents fundamental characteristics related to the manufacturability, and can be applied in 3D printing processes, which allows the construction of structures with more complex geometries that can be efficient, yet has a high specific weight and not confers a high resistance like CFRP, but depending on the structure, the PLA can bring rigidity to it.

The core, in turn, assumes the role of increasing the cross-section of the frame structure as a whole. Thus, by applying a core that is capable of significantly increasing the cross-section and increasing the moment of inertia [4] of the same, allowing better performance of the structure, in addition to being able to withstand greater loads. In addition, the core, despite supporting significantly smaller efforts than those undergone by the reinforcement, should be made of a material with high shear strength, allowing to accommodate this property, to the characteristics already presented by the reinforcement. The core also allows a greater facility for the laminate to have more favorable geometric characteristics, as is the case of the mentioned cross section.

Property	Unit	H80 Foam	lite ply
Tensile Strength	MPa	2.5	31.05
Compressive Strength	MPa	1.4	36.2
Young Modulus	GPa	0.09	9.3
In-Plane Shear Strength	MPa	1.15	1.90
In-Plane Shear Modulus	MPa	27	318.9
Density	kg/m^3	80	500

Table 4: Mechanical Properties of the analyzed structural materials

Among the materials analyzed, the manufacturability of the material must be reconsidered, as well as the necessary characteristics for the design of the MAV, which needs to be light and compact. PVC H80 foam clearly has a much lower specific weight and is also easy to manufacture. The H80 foam also allows high stiffness due to its mechanical properties. lite ply also gives high rigidity when laminated with CFRP, however, because it is commercialized in boards with a limited thickness of 3 mm, it does not have high versatility when designing and constructing frames with more complex geometries, as in the case of laminated H80 foam with CFRP.

Through the selected materials, various configurations were analyzed as the composite materials are able to bring different mechanical properties depending on the composition and arrangement of the core and reinforcement. Thus, different configurations for the drone structure were studied. Figure 4 shows four configurations that were studied. The materials arrangements of each configuration are shown in Table 5.

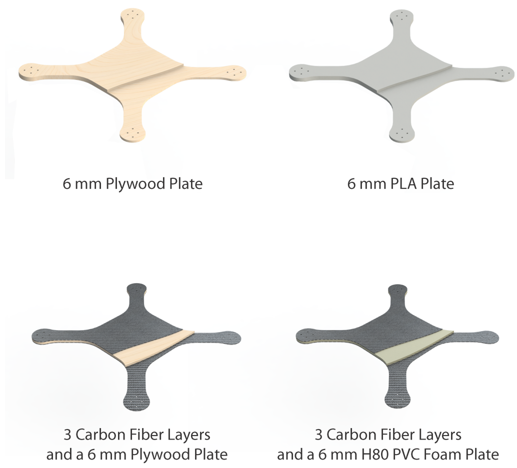


Figure 4: Possible Material Composition of the Frame

Configuration	Core Material	Reinforcement Material
1	H80	CFRP
2	Lite ply	CFRP
3	-	Lite ply
4	-	PLA

Table 5: Material of the frames analysed

5 Simulation

To evaluate the performance of the four configurations, all types were modeled in Solidwoks and both static and dynamic finite element analysis (FEA) were performed for each one. All analysis were made using Solidworks simulation tool, tetrahedral elements were used during mesh creation. Thus, An adaptive mesh was also used during meshing for better results. Each simulation was repeated, by increasing the number of elements until the convergence.

To ensure good stability and control of the MAV, the frame must withstand the flight loads without large displacements. The FEA static analysis was made in order to evaluate the maximum displacement of the frame under the design loads shown in Table 1. The boundary condition defined for this analysis was the center plate as a fixed area, representing the local of the heavier components. Loads, as motor forces and landing impact, were defined in the proper position. The displacements of configuration 1 are shown in Figure 5.

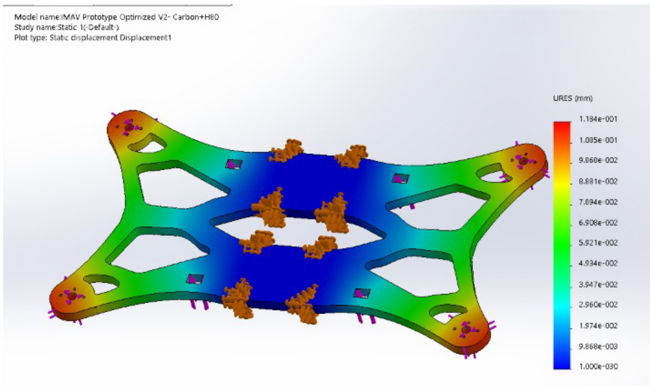


Figure 5: Static Analysis in Configuration 1

A modal analysis was made to estimate the natural frequency of the first mode. The final design should has its first natural frequency greater than $150Hz$, for dynamic stability of the structure during flight. To perform the analysis some assumptions were defined. First, the frame was considered in free vibration, then, no boundary condition constraints were defined, which simulates the flight condition. MAV components were simplified as concentrated masses in their position on the frame. Figure 6 presents the first elastic mode shape of configuration 1.

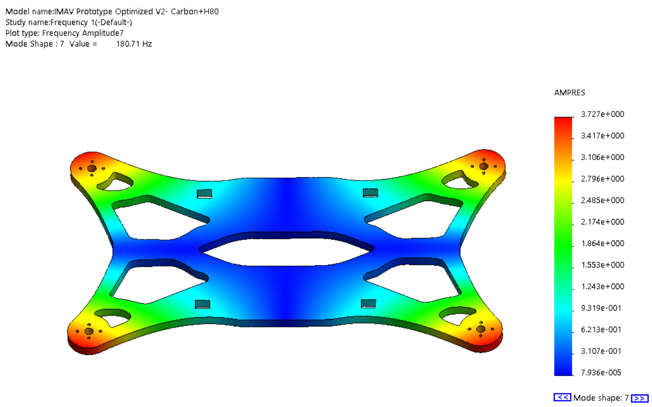


Figure 6: First elastic mode shape of configuration 1

The maximum static displacement, first natural frequency and weight were estimated and compared for each configuration. Thus, in the Table 6 are the data of the finite element simulations, which indicate the optimal configuration that combines all the aspects analyzed here.

Conf.	**Max. Static Displacement (mm)**	**First Mode Frequency (Hz)**	**Structure Mass (g)**
1	0.1184	180.71	66
2	0.0442	285.08	130
3	0.3125	106.55	76
4	1.031	52.40	197

Table 6: Configurations Analysis

6 MANUFACTURING

A vacuum bagging lay-up technique is used to manufacture CFRP. The technique removes the excess of resin, which is mostly applied to achieve higher carbon fiber concentrations and, consequently, higher mechanical properties.

A PVC H80 foam was chosen as the structure sandwich core. The propose process aims at improving the moment of inertia of the drone arms, by increasing the distance between bottom and top carbon fiber layers. The lay-up is made directly on a square foam core, on both sides of the plate, which eliminates the need and machining of hard material molds.

During the lay-up, an epoxy resin of the same weight of carbon fiber is mixed to wet the fabric. Besides, a breather and a perforated film are used between the vacuum bag and the fabric to absorb excess of epoxy. The result of this process is a 8 mm carbon-foam sandwich plate with 1 mm CFRP laminate with approximately 35% of epoxy resin and 65% of carbon fiber in weight, at a vacuum of $-600mmHg$. Figure 7 shows the first step of the lay-up procedure.

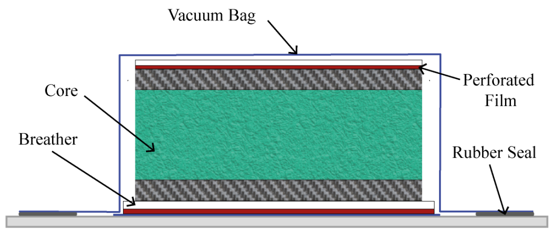


Figure 7: Lay-up technique of the composite frame

After the lay-up process, a CNC (Computer Numerical Control) milling machine is used to mill the sandwich plate to the desired design. A CNC machine is important to precisely achieve the layout optimized by the methodology, which insures the expected weight relief. Besides, the selected materials allow a small machine time and the durability of the milling tool.

7 Results

After the results of the topological optimization, finite element simulations and material selection, it was possible to obtain complete analysis of the frames in the previously mentioned configurations. The simulation results obtained for configuration 1 are highlighted in Figures 6 and 5.

As shown above, the frames modeled with the geometry indicated by the topological optimization, show how the frames behave to the conditions originally established by the optimization. The PLA frame clearly exhibits poor overall performance compared to the other configurations shown and has a very high overall weight for the limited thrust of the motors. However, the lite ply frame only has a low total structural weight and is relatively interesting for the considered dimensions, however, the static simulations indicate that the maximum deformation of the frame is still very high when compared to the composite sandwich frames. Within composite frames, there are two analyzes that can be done. The configuration model 1 presents good resistance results, however, in the frequency analysis simulation, it ended up not having a rigidity, due to the foam in the core. However, the model of Configuration 2 represents the one that best behaves statically and in the analysis of frequencies, to the weight of 136 g which is significantly high considering that the sum of the masses of the electronic components chosen turns around 350 g. Thus, the model of Configuration 1 represents the ideal model, since it presents high structural efficiency and stiffness. Configuration 2 would be ideal considering a selection of a different propulsion system that supports a larger structural weight.

Figure 8: Final result of the MAV frame

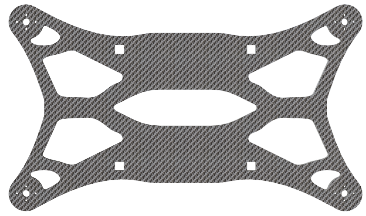


Figure 9: Top view of the optimized geometry

8 Conclusion

In this paper, authors present the development of a optimized MAV frame, which involves several aspects. In the context of the IMAV 2018 indoor competition, the development of light and compact frames involves structural characteristics such as strength and stiffness. Thus, the topological optimization allowed the study of complex geometries, reaching improved structural properties. By minimizing compliance, it was possible to obtain an optimized geometry for the loads considered here. Through the analysis of materials, four different frame combinations were performed by FEA simulations, combining materials such as CFRP, PVC foam and Lite ply. These analysis were focused on static and modal simulations and aimed at validating the geometry obtained in the topological optimization. Through the results, an final configuration was obtained considering the disposition of materials and also their manufacturing process. The designed frame of $66g$ follows the defined constraints parameters, as displacement and first natural frequency, allowing a high score in the mass multiplier parameter in the competition.

Acknowledgements

We would like to thank the Brazilian funding agencies CNPq, FAPERJ and CAPES for continued support and supplied resources.

References

[1] Ji-Hong Zhu, Wei-Hong Zhang, and Liang Xia. Topology optimization in aircraft and aerospace structures design. *Archives of Computational Methods in Engineering*, 23(4):595–622, 2016.

[2] Martin Philip Bendsøe and Noboru Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer methods in applied mechanics and engineering*, 71(2):197–224, 1988.

[3] Ole Sigmund. A 99 line topology optimization code written in matlab. *Structural and multidisciplinary optimization*, 21(2):120–127, 2001.

[4] James M Gere and Barry J Goodno. Mechanics of materials 5th. *Brooks Cole*, page 780, 2001.

Disturbance observer based Control Design of Flapping-wing MAV Considering Model Uncertainties

Inrae Kim[1], Heetae Park[1], Seungkeun Kim[1]*, Jinyoung Suk[1] and Sungbong Cho[2]
[1]Department of Aerospace Engineering, Chungnam National University, Daejeon, 34134, Republic of Korea
[2]Agency for Defence Development, Daejeon, 34186, Republic of Korea

ABSTRACT

This paper presents controller design for the stable flight of Flapping-Wing Micro Aerial Vehicle (FWMAV). FWMAV has model uncertainty in aerodynamics and vehicle dynamics generated by the flapping wing motions. Hence, for the robust controller design, disturbance observer-based controller (DOBC) is designed considering these model uncertainties as system disturbance. Dynamic modeling takes into account flapping wing kinematics, aerodynamics, and 6DoF dynamics. Finally, to verify the performance of the applied controller, numerical simulations are performed in a MATLAB environment in hovering flight condition, and the result of disturbance observer based controller is analyzed.

1 INTRODUCTION

In recent decades, unmanned aerial vehicle (UAV) have experienced rapid development. Micro air vehicle (MAV) is a class of UAVs restricted in size and has been tackled through biomimetic technology mimicking insects [1, 2, 3]. This is because insects can be highly maneuverable and have stable hovering and excellent aerodynamic characteristics unlike fixed-wing and rotary-wing aircraft. This feature can effectively help MAVs perform a variety of missions, including surveillance, reconnaissance, secret infiltration, and search inside a collapsed building. Also, it would be advantageous to swarm flight in the future.

Especially, aerodynamic analysis for flapping in the area of low Reynolds number, actuator, micro sensor, battery, etc, the research are still studying for micro miniaturization in the actual making as well as the theoretical and analytical aspects. For most of the MAV's study, it was carrying out experiments by using the external power supply. State variables of the MAV are measured from the outside because of the weight limit. Then, the measured value was used to study the attitude control from the outside [4]. The attitude control of MAV is very necessary for excellent performance. So, a lot of simulation was conducted for attitude control over the world.

*Corresponding Author: skim78@cnu.ac.kr

In Ref. [5], the study for a variety of input control was conducted about flapping frequency or phase using pitch attitude change by flapping motion. Ref. [6] presented key aspects of active flexible wing technique. Cheng and Deng [7] proposed a mathematical model of near-hover attitude dynamics and control in flapping flight. Study of insect mimicking is more focus on modeling than control. But, these MAV attitude control is essential to basic techniques for the design and production and system development in future. Many people have been trying to solve control problems in a variety of fields, so that more advanced control design techniques was announced as a result of these efforts. However, despite these recent research, actually, PID controller is still the most. Among other reasons, because PID is relatively easy to design a controller, good and economic effects. Disturbance observer for similar reasons has been used much since the late 1980s. The DOB scheme which is a kind of more straightforward and efficient disturbance suppression mechanism is developed by [8] To date, research was conducted about DOB. There is a study that applies to difficult non-linear and complex system. Nevertheless, a strong disturbance removal performance shows a simple feature. The DOB control system had been successfully applied to many fields and proved to be an effective countermeasure against the disturbance. Through these advantages, apply to FWMAV.

The overall structure of this paper is given as follows. Section 2 introduces a mathematical model of FWMAV, which combines quasi-steady aerodynamics and rigid body dynamics. Section 3 proposes the flapping flight control based on the PID controller and Track guidance, and additional disturbance observer based control. Section 4 presents a numerical simulation environment and results of FWMAV waypoint flight. Last, conclusions and future work are given in section 5.

2 MATHEMATICAL MODELING

2.1 Coordinate system

This paper uses four coordinate frames to describe flapping kinematics and body dynamics as shown in Fig. 1: 1) E : Earth-fixed frame, a fixed coordinate system on the Earth's surface; 2) B : Body-fixed frame, fixed coordinates to the center of gravity for a vehicle; 3) S : Stroke-plane frame, fixed coordinates to each connecting point between each wing and a body; and 4) W : Wing-fixed frame, fixed coordinates to the center of pressure for each wing.

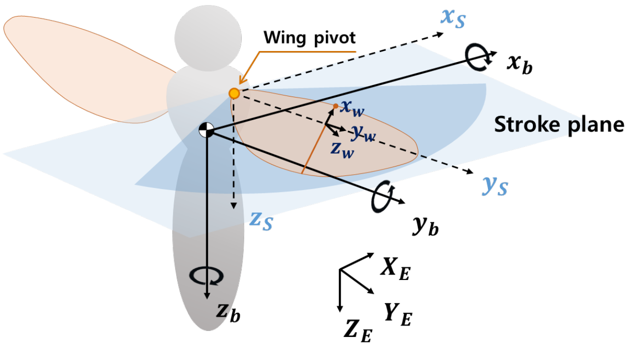


Figure 1: Coordinate system of FWMAV

2.2 *Flapping wing kinematics*

The motion of the flapping wing can be defined with three independent angular degrees of freedom at the wing base pivot as shown in 'Figure 3': 1) ϕ : Stroke angle, back and forth motion of the wing; 2) θ : Deviation angle, up and down motion of the wing with respect to the stroke plane; and 3) ψ : Feathering angle, the geometrical angle of attack of the wing surface.

FWMAV is flying by the force generated by the flapping motion. Actual insect flapping motion has different complicated motion depending on the type and size of insects [9]. In fact, it is difficult to apply real insects flapping motion to FWMAV. Therefore, this paper considers simple flapping. Based on this wing kinematics definition, the motion of each rotational degree of freedom is defined with a sinusoidal function, to reproduce the periodic wing beat motion of the flapping flight, 'Equation 1 to 4' show the wing kinematics function for each rotational degree of freedom :

$$\phi(t) = \phi_{amp} \cos(2\pi f t) + \bar{\phi} \tag{1}$$

$$\psi(t) = \frac{\psi_{amp}}{\tanh(C_\alpha)} \tanh\left(C_\alpha \sin(2\pi f t) + \frac{\pi}{2}\right) + \bar{\psi} \tag{2}$$

$$\alpha(t) = \frac{\pi}{2} - sign(\dot{\phi})\psi \tag{3}$$

$$\theta(t) = 0 \tag{4}$$

where t is time, f is the typical flapping frequency of FWMAV, and C_α is a coefficient for adjusting the interval of the stroke reversal. Each kinematics function has two variables, which are the amplitude and bias. The angle of attack(α) can be expressed as 'Equation 3' and the value of the angle of attack is determined according to the sign of $\dot{\phi}$.

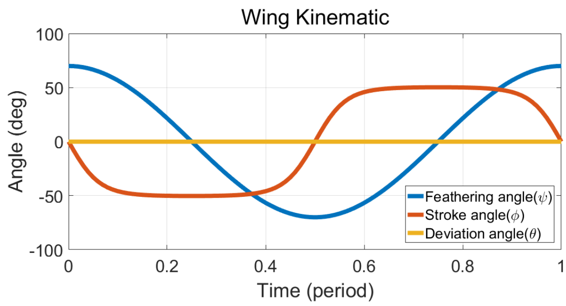


Figure 2: Amplitude of the wing kinematics

2.3 *Flapping flight aerodynamics*

The aerodynamic forces generated by the wings are complex and the motion changes rapidly in low Reynolds number flows. In the absence of skin friction, the instantaneous forces generated by a thin, flapping wing may be represented as the sum of four force components, each acting normal to the wing surface :

$$F_{total} = F_{trans} + F_{rot} + F_{add} + F_{wc} \tag{5}$$

where F_{trans} is the instantaneous translational force, F_{rot} is the rotational force, F_{add} is the force due to the inertia of the added mass of the fluid and F_{wc} is the force due to wake capture. Among the four major aerodynamic components, the aerodynamics generated from a translational and a rotational motion of the wing is modeled for this study [10, 11]. The wake capture effect is neglected because the inherent unsteadiness of the phenomenon makes it hard to be modeled with a quasi-steady approach. The instantaneous translational and rotational forces from each aerodynamic strip are calculated by

$$\begin{aligned} F_{tr,T} &= \tfrac{1}{2}\rho A_w C_T U_{cp}^2 \\ F_{tr,N} &= \tfrac{1}{2}\rho A_w C_N U_{cp}^2 \end{aligned} \tag{6}$$

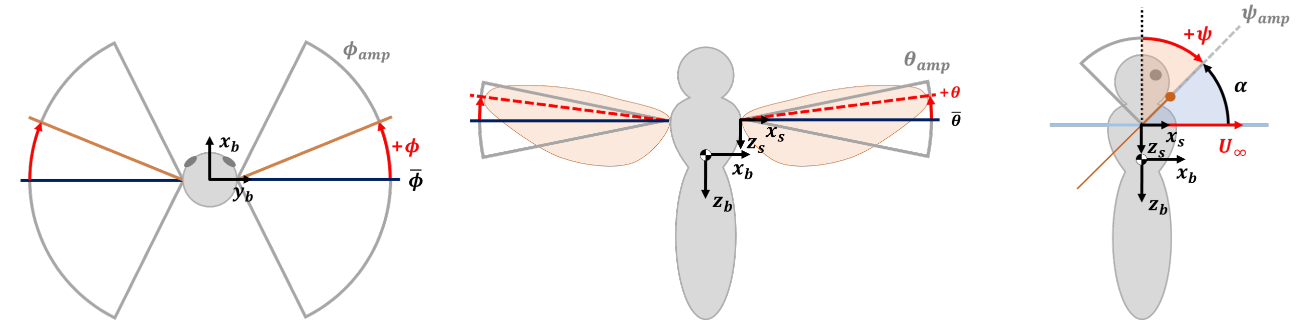
where U_{cp} is the velocity of the wing at the center of pressure. In small advance ratio, this parameter can be defined as :

$$U_{cp} = \hat{r_2} L \dot{\phi} \tag{7}$$

In which $\hat{r_2}$ is the normalized distance of center of pressure from wing base :

$$\hat{r_2}^2 = \frac{\int_0^L c(r) r^2 dr}{L^2 A_w} \tag{8}$$

where ρ is the air density, A_w is the wing area, L us the wing length, $\dot{psi}$ is stroke velocity, and c is wing chord width. C_N and C_T are the force coefficient each Normal, tangential. This force coefficient for the model wing was measured in [3] and fitted with the following equation :



(a) Stroke angle ϕ
(top view)

(b) Deviation angle θ
(back view)

(c) Feathering angle ψ
(side view)

Figure 3: Wing Kinematics

$$
\begin{aligned}
C_N &= 3.4 \sin\alpha \\
C_T &= \begin{cases} 0.4\cos^2(2\alpha) & 0^\circ < \alpha < 45^\circ \\ 0 & otherwise \end{cases}
\end{aligned} \tag{9}
$$

α stands for the angle of attack. A quasi-steady treatment of the aerodynamic force due to wing rotation, $F_{rot,N}$, was calculated by [12] :

$$
F_{rot,N} = \frac{1}{2}\rho A_w C_{rot} \hat{c} c_m \dot{\alpha} U_{cp} \tag{10}
$$

where C_{rot} is the theoretical value of rotational coefficient, $\hat{c} = c/\bar{c}$ is the normalized chord length, and c_m is the maximum wing chord width. then C_{rot} and $\hat{c}$ is given by :

$$
C_{rot} = 2\pi(0.75 - \hat{x_0}) \tag{11}
$$

$$
\hat{c} = \frac{\int_0^L c^2(r) r dr}{\hat{r_2} L A_w c_m} \tag{12}
$$

$\hat{x_0}$ is the dimensionless distance of the longitudinal rotation axis from the leading edge. Therefore, the total lift and drag forces are computed as follow :

$$
\begin{aligned}
F_N &= F_{tr,N} + F_{rot,N} \\
F_T &= F_{tr,N}
\end{aligned} \tag{13}
$$

$$
\begin{aligned}
F_L &= F_N \cos\alpha + F_T \sin\alpha \\
F_D &= -F_N \sin\alpha - F_T \cos\alpha
\end{aligned} \tag{14}
$$

2.4 *Body dynamics*

The dynamics of the FWMAV can be described, under rigid body assumption, by Newton-Euler motion equations. Similar to an aircraft we obtain 12 ordinary differential equations with 12 unknown coordinates.

Force equations.

$$
\begin{aligned}
\dot{u} &= rv - qw + g_x + a_x \\
\dot{v} &= pw - ru + g_y + a_y \\
\dot{w} &= qu - pv + g_z + a_z
\end{aligned} \tag{15}
$$

where

$$
\begin{aligned}
g_x &= -g \sin\theta \\
g_y &= g \sin\phi \cos\theta \\
g_z &= g \cos\phi \cos\theta
\end{aligned} \tag{16}
$$

Here, $[u, v, w]$ denotes the body-axis velocities, $[p, q, r]$ indicates the body-axis angular velocities, $[\phi, \theta, \psi]$ is the Euler angles from navigation to body frame, and g is the acceleration of gravity. Moreover, $[a_x, a_y, a_z]$ include all the airframe loads from aerodynamics, propulsion, and winds [13].

Moment equations.

$$
\begin{aligned}
\dot{p} &= (c_1 r + c_2 p) q + c_3 L + c_4 N \\
\dot{q} &= c_5 pr - c_6(p^2 - r^2) + c_7 M \\
\dot{r} &= (c_8 p - c_2 r) q + c_4 L + c_9 N
\end{aligned} \tag{17}
$$

where $[L, M, N]$ is moments with respect to the center of gravity. We define inertia coefficient c_1 to c_9 as 'APPENDIX A'

$$
\begin{bmatrix} mI & 0 \\ 0 & I_b \end{bmatrix} \begin{bmatrix} \dot{v}_b \\ \dot{w}_b \end{bmatrix} + \begin{bmatrix} w_b \times m v_b \\ w_b \times I_b w_b \end{bmatrix} = \begin{bmatrix} \Sigma F \\ \Sigma M \end{bmatrix} \tag{18}
$$

where m is the mass of the FWMAV, I_b is the inertia matrix relative to the center of mass, I is the the 3×3 identity matrix, and v_b and w_b are the linear and angular velocity vectors in body frame coordinates [3, 14]. Since the lift and drag forces are given by 'Equation14' are calculated relative to the Stroke-plane frame, a coordinate transformation is necessary before obtaining the forces and torques acting on the body frame.

$$
\begin{bmatrix} F_a \\ M_a \end{bmatrix} = R_{sb} \begin{bmatrix} F_{a,S} \\ M_{a,S} \end{bmatrix} \tag{19}
$$

where R_{sb} is the rotation matrix of the body frame relative to the stroke plane, $F_{a,S}$, and $M_{a,S}$ are the aerodynamic force and moment generated in the stroke-plane frame and F_a and M_a are the aerodynamic force and moment in the body frame. Given the lift and drag generated by aerodynamics, together with the stroke angle, the forces and torques in the stroke plane can be calculated as

$$F_{a,S} = \begin{bmatrix} -F_{D,l}\cos\phi_l - F_{D,r}\cos\phi_r \\ -F_{D,l}\sin\phi_l + F_{D,r}\sin\phi_r \\ -F_{L,l} + F_{L,r} \end{bmatrix} \tag{20}$$

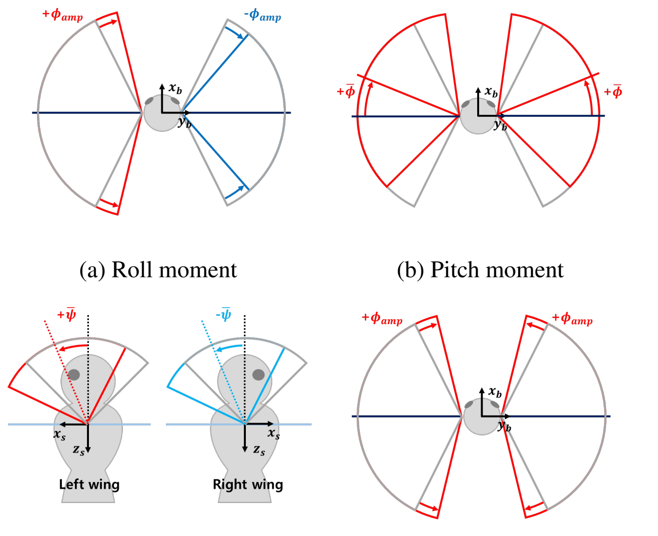
$$M_{a,S} = \hat{r_2}L \begin{bmatrix} F_{L,l}\cos\phi_l - F_{L,r}\cos\phi_r \\ F_{L,l}\sin\phi_l + F_{L,r}\sin\phi_r \\ -F_{D,l} + F_{D,r} \end{bmatrix} \tag{21}$$

These forces and moments are time-varying nonlinear equations for the flapping motion.

3 Flapping flight control

3.1 Control parameters

To perform attitude and position control, force and moment must be generated for each axis. Many universities abroad have generated moments in different ways and have chosen appropriate forms for each study [1, 15]. In this study, the force and moment generation method that can implement FWMAV is selected from among them and it is shown in 'Figure 4' [16].



(a) Roll moment (b) Pitch moment

(c) Yaw moment (d) Z-axis force

Figure 4: Control force and moments generated through wing kinematics parameters

a) Roll moment is generated by giving a difference to the amplitude of the stroke angle of both wings; b) Pitch moment is generated by moving the center line of the stroke angle back and forth; c) Yaw moment is generated by moving the center line of the feathering angle in the opposite direction; , and d) Z-axis force is generated by giving a same amplitude of the stroke angle.

$$\phi(t) = (\phi_{amp} + \Delta\phi_{roll} + \phi_{hover})\cos(2\pi f t) + \phi_{pitch} + \bar{\phi} \tag{22}$$

$$\psi(t) = \frac{\psi_{amp}}{\tanh(C_\alpha)}\tanh\left(C_\alpha \sin(2\pi f t) + \frac{\pi}{2}\right) + \Delta\psi_{yaw} + \bar{\psi} \tag{23}$$

FWMAV use control inputs through flapping motion. 'Equation 22' is the Amplitude of the stroke angle is a control input variable and is used both as altitude, roll and pitch control. The flapping amplitude of one wing can be up to 180°. Therefore, the sum of the flapping amplitude control inputs must be limited to not exceed ±90°. The feathering angle is a control input variable for yaw control. The limit of this variable is determined by the forces that occur. In consideration of this, as shown in 'Table 1'.

Effect	Variable	Range
M_x	$\Delta\phi_{amp}$	$[-5°5°]$
M_y	$\bar{\phi}$	$[-5°5°]$
M_z	$\Delta\bar{\psi}$	$[-1°1°]$
F_z	ϕ_{amp}	$[-10°10°]$

Table 1: Range values of each control variable.

3.2 Attitude and position control

In this paper, PID(Proportional-Integral-Derivative) control is used for attitude and position control of the FWMAV. PID control is a classical automatic control technique that is most widely used in industrial control applications. Its design, implementation and tuning are simple since the design itself are response-based, it can be designed without knowledge of the system. In this study, a basic flight control system is designed as a multiple cascaded PID controller as shown in 'Figure 5'. Attitude controller is designed as an angular position and rate cascaded P-PID inner-outer loop structure to provide more stability and faster response due to rate feedback.

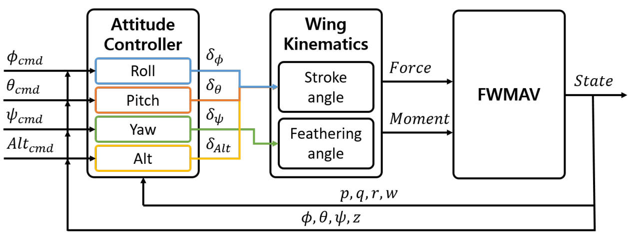


Figure 5: Schematic of the basic flight control system

3.3 Track Guidance

This section proposes a track guidance algorithm for efficient waypoint navigation of FWMAV. this paper assumes FWMAV flight conditions. 1) FWMAV is capable of hovering, but in case of waypoint flight, it makes a continuous flight with constant advance speed. 2) The heading of the vehicle coincides with the forward direction. And, track guidnance system is designed as shown in 'Figure 6'.

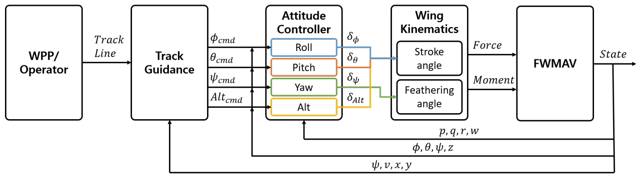


Figure 6: Structure of the track-guidance algorithm for FWMAV

When the waypoint of the FWMAV is determined, the algorithm of the track guidance has the following flow.

ALGORITHM

Input last and current waypoint position data

1: Caulculate the straight-line direction vector between the waypoints. (Track Heading)
2: Command the vehicle heading to match the Track heading. (ψ_{cmd})
3: Calculation of Cross track error Distance between Track and vehicle.
4: Determine the Cross track error sign code by the sign of ψ_d.
5: Instruction ϕ_{cmd} so that the value of Cross track error is 0.
6: When a vehicle enters the Waypoint Range, it moves to the next waypoint

First, calculate the heading angle from the previous waypoint to the current waypoint. This is called Track Heading. Next, ψ_{cmd} is created to match the heading of the FWMAV with the tracking heading. In this case, the heading is aligned with the Track direction, and then the command is generated to enter the Track. Since we know vehicle position and waypoint, we can calculate the Cross-track error distance, which is the vertical distance between vehicle and Track. At this time, the Cross-track error distance does not include sign information in the Track. Therefore, it is necessary to check whether the FWMAV is on the left or right side of the Track. In this paper, we calculate the angle between the track heading obtained above and the current waypoint. At this point, if ψ_d is positive, there is FWMAV on the right side of the Track if it is negative on the left side of the Track. After calculating distance and sign, ϕ_{cmd} is generated so that Cross track error becomes 0. Then, when the aircraft enters the predetermined waypoint range, it moves to the next waypoint.

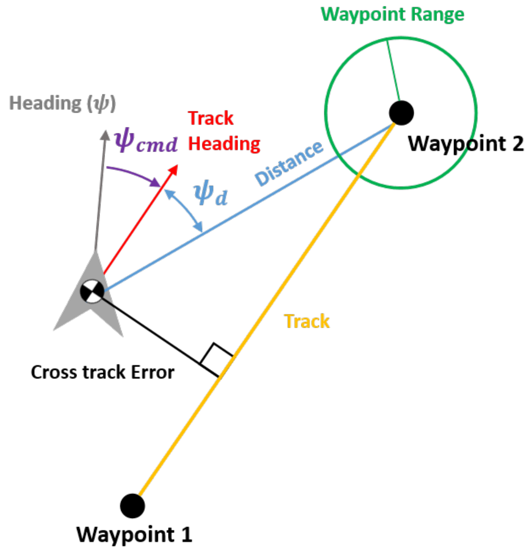


Figure 7: Track guidance algorithm for FWMAV

3.4 Disturbance observer based control

Disturbance generally exists in control systems and degrades performance or stability of the closed-loop system. Especially, aerial systems are subject to forces and moments caused by external wind and unmodeled dynamics due to complex aerodynamics. Thus, disturbance rejection is the major consideration for the control design of the aerial system to achieve safe flight. Disturbance observer implies various control technique that estimates disturbance in a broad sense. This study, Disturbance observer that uses the inverse of a nominal model is considered. This disturbance observer is an additional controller that can be designed separately from the outer loop controller design. In the concept of the disturbance observer, the disturbance does not only refer to the disturbance from the external environment but also uncertainties from the plant including unmodeled dynamics, parameter perturbations, and nonlinear couplings as a lumped disturbance.

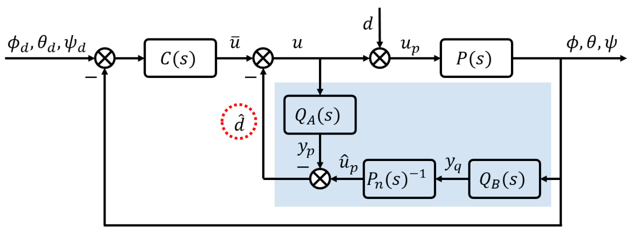


Figure 8: Disturbance observer control structure

Disturbance observer based control structure is shown in 'Figure 8'. $P(s)$ is the actual system with uncertainties,

ϕ_d,θ_d,ψ_d and u_p is input and output of the system, respectively. $C(s)$ is the outer loop controller. It is difficult to know the actual system $P(s)$ perfectly. However, the nominal model $P_n(s)$ can be seen. Therefore, if we know the output of the model and the control input, we can estimate the disturbance and compensate the disturbance controller. The implementation of Q-filter is to filter the noise and make an appropriate transfer function of $P_n(s)^{-1}$. The concept of the disturbance observer is to estimate the input to the system including disturbance (u_p) using the inverse of the nominal model($P_n(s)^{-1}$), and subtract the outer controller output (y_p); then it becomes the estimated disturbance ($\hat{d}$) as follows:

$$u = \bar{u} + y_p - \hat{u}_p = \bar{u} - \hat{d} \tag{24}$$

The desired dynamics of the system is considered as inertia-moment dynamics without friction. For example, Taking lateral dynamics can be simplified as follows :

$$L = J_{xx}\ddot{\phi} \tag{25}$$

$$L = c_t\delta_\phi - (J_{zz} - Jyy)\dot{\theta}\dot{\psi} \tag{26}$$

$$P_n(s) = \frac{\phi(s)}{\delta_\phi(s)} = \frac{1}{J_{xx}s^2} \tag{27}$$

Since a relative degree of the nominal plant is two, the minimum required the relative degree of the Q-filter is also two as:

$$Q_A(s) = Q_B(s) = \frac{a_0/\tau^2}{s^2 + (a_1/\tau)\,s + (a_0/\tau^2)} \tag{28}$$

where the coefficients (a_0 and a_1) of the Q-filter are chosen so that the following two expressions have negative real parts and that τ is small enough [17, 18].

4 Numerical simulation

4.1 Simulation environment

This section carries out numerical simulations using the proposed controller, track guidance, and disturbance observer based control. The simulation works in MATLAB's Simulink. The simulation initial values were modeled by hummingbirds similar in size to the flapping-wing MAV with reference to various papers [19]. The detailed specifications are summarized in 'Table 2'.

The Simulation was performed under three conditions to verify the performance of the designed controller. At this time, the initial Euler angle and initial position in each environment are [0 0 0]. 1) Fly a diamond path in a non-disturbance environment. 2) Fly a diamond path without a disturbance observer in an environment where the north wind is blowing. 3) Diamond phth flight when a disturbance observer is used in a borehole environment.

Parameter	Value	Unit
ρ	1.225	kg/m^3
g	9.81	m/s^2
M	4.32e-3	kg
I_{xx}	4.92e-7	-
I_{yy}	5.57e-7	-
I_{zz}	4.11e-7	-
I_{xz}	2.2e-7	-
L	4.8e-2	m
$\hat{r_2}$	0.6	-
$\hat{x_0}$	0.25	-
c_m	1.9e-2	m
$\hat{c}$	0.6	-
A_w	6.11e-4	m^2
f	50	Hz
C_α	2.5	-
ϕ_{amp}	70	deg
ψ_{amp}	50	deg
$\bar{\phi}$	-3	deg
$\bar{\psi}$	0	deg

Table 2: Main parameter for simulation.

4.2 Simulation result

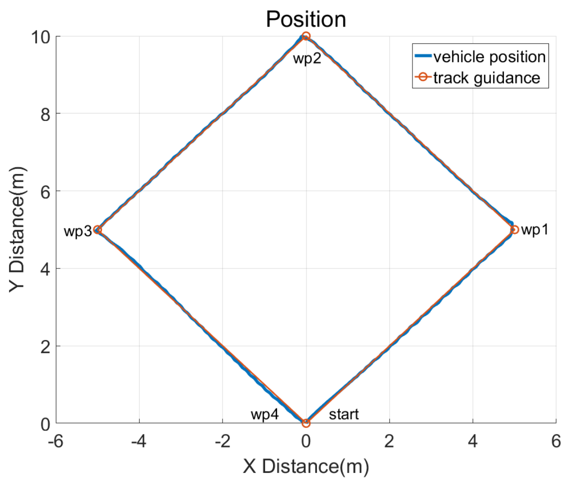
1. Non disturbance

'Figure 9a' shows the flight path of FWMAV. In the environment without disturbance, it can be confirmed that it follows well along the waypoint. Detailed position information can be found 'Figure 9b'. For X,Y position, the command value follows well. when the FWMAV reached the Waypoint range, the command changes to the next waypoint, and the algorithm works well. The Z position will overshoot but will follow the command value well. 'Figure 9c' compares the Euler angle command value with the vehicle Euler angle. It can be confirmed that ψ first follows the command value and becomes stable. θ has a constant value except when the waypoint is changed since the command value is generated so as to have a constant advancing direction. ϕ is repeatedly commanded to reduce the cross track error, and a "zig-zag" motion is generated by chattering around the track.

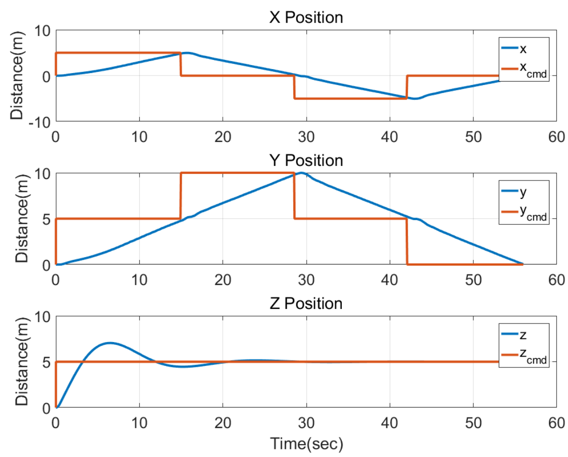
2. disturbance

This simulation includes disturbances. The disturbance assumed that the north wind was divided by a constant wind. The graph for the disturbance is shown in 'Figure 11'. The simulation environment is the same except for disturbance.

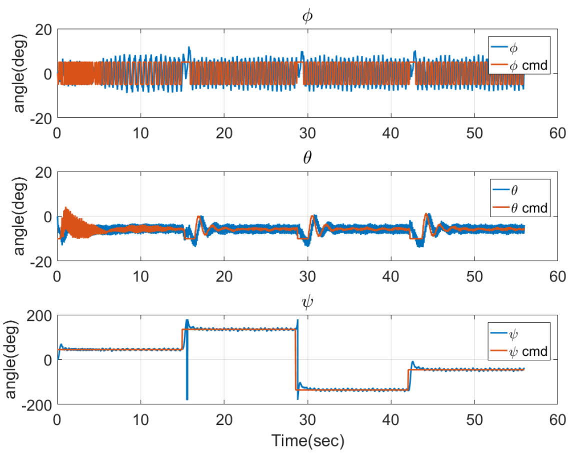
'Figure 11a' shows the flight path of the FWMAV in a disturbed environment. Compared to 'Figure 9a', it can be



(a) FWMAV Path



(b) Position command



(c) Euler angle command

Figure 9: Simulation of non-disturbance

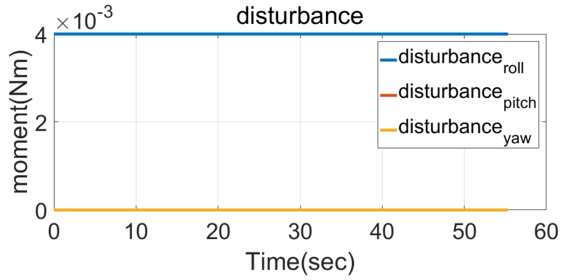
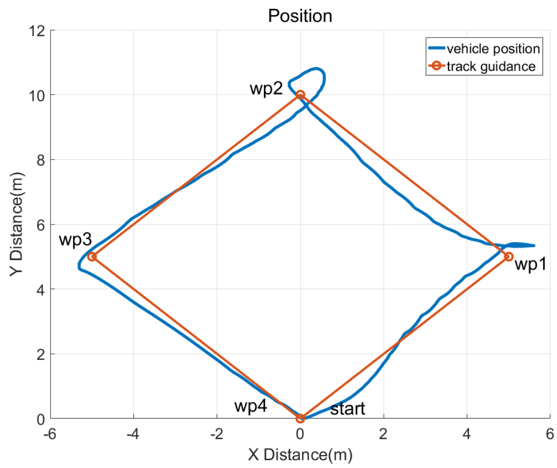
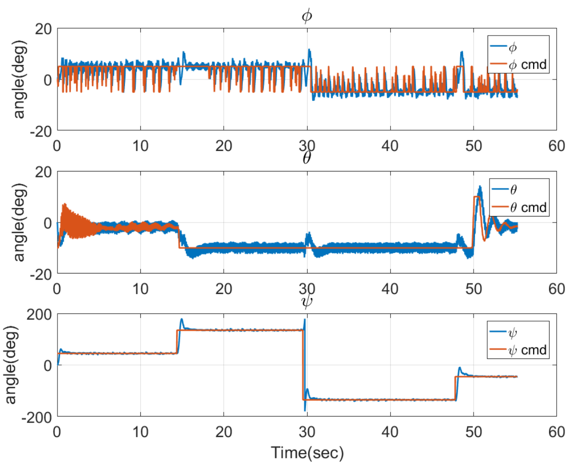


Figure 10: Disturbance

seen that it deviates much from the track, and it is far from the track around waypoint 1. The Euler angle can be confirmed by the fact that the track response is pushed by the wind and the command value response of ϕ and θ changes. In the case of ψ, the path of the track is constant, which is the same as when there is no disturbance.



(a) FWMAV Path

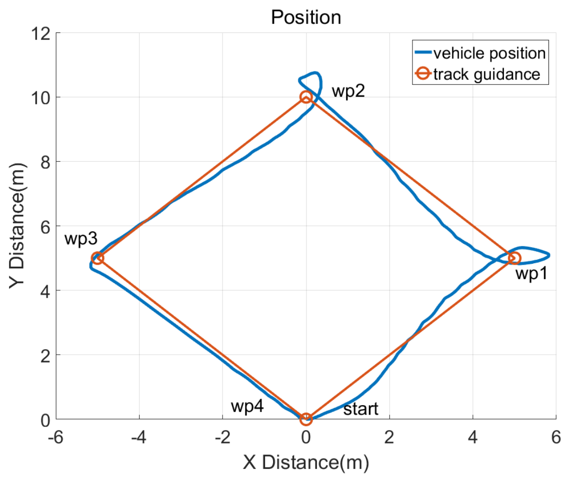


(b) Euler angle command

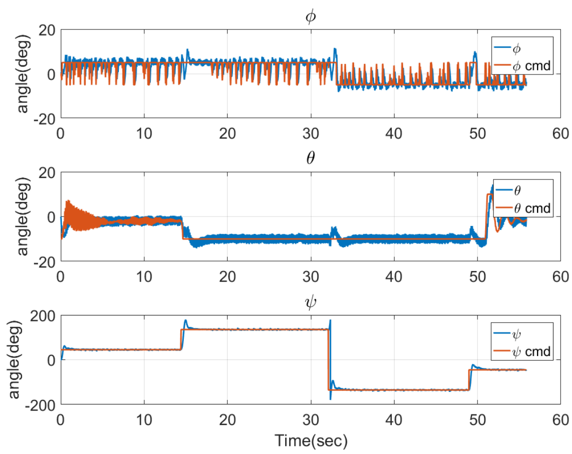
Figure 11: Simulation of disturbance

3. disturbance using DOBC

'Figure 12' is shows the simulation results using DOBC. 'Figure 12a' shows the flight path of the FWMAV, followed by a little more track than without the DOBC. Euler angle is similar to that of the absence of the DOBC, but it can be seen that the FWMAV's Euler angle is more stable. and the amplitude of the Euler angle is lowered and stabilized in the sections where overshoot occurs.



(a) FWMAV Path



(b) Euler angle command

Figure 12: Simulation of disturbance with DOBC

5 Conclusion

In this paper, numerical modeling of the FWMAV is performed, implemented control variables, controller, and track guidance algorithm are designed. Numerical simulations were performed to verify the performance of waypoint flight in an environment with disturbance. Simulation results show that the FWMAV performed stable flight for a given waypoint in a non-disturbance environment. On the other hand, in the given environment of constant disturbance, it showed flight off the track, but the waypoint flight could be completed. We designed a disturbance observer to simulate a stable flight in a disturbance environment. As a result, the disturbance observer was able to follow the path well, but it showed a slight effect. Rather, it showed a good effect on stabilization of FWMAV's Euler angle. Simulation results show that there is a need to find a way to eliminate the chattering effect of the track guidance algorithm. As a result, unnecessary control command values are included, which can impair the control actuator in the real vehicle. In addition, the problem that the effect of the disturbance observer designed for stable flight in the disturbance environment was small will be complemented by the redesign of the disturbance observer. After completing the current simulation, we plan to model the actuator and sensor of the actual FWMAV in order to construct a simulated environment similar to the actual vehicle, and apply it to the simulation and analysis.

Acknowledgements

This research was supported by a grant to Bio-Mimetic Robot Research Center funded by Agency for Defense Development and Defense Acquisition Program Administration. (UD130070ID) and the research project (10062327) funded by the Ministry of Trade, Industry and Energy of Korea, Republic of.

References

[1] Keennon, Matthew, Karl Klingebiel, and Henry Won. "Development of the nano hummingbird: A tailless flapping wing micro air vehicle." 50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition. 2012.

[2] Ma, Kevin Y., et al. "Controlled flight of a biologically inspired, insect-scale robot." Science 340.6132 (2013): 603-607.

[3] Deng, Xinyan, et al. "Flapping flight for biomimetic robotic insects: Part I-system modeling." IEEE Transactions on Robotics 22.4 (2006): 776-788.

[4] Fuller, Sawyer B., et al. "Controlling free flight of a robotic fly using an onboard vision sensor inspired by insect ocelli." Journal of The Royal Society Interface 11.97 (2014): 20140281.

[5] Loh, Kum Hoe, and Mike Cook. "Flight Dynamic Modelling and Control System Design for a Flapping Wing Micro Aerial Vehicle at Hover." AIAA Atmospheric Flight Mechanics Conference and Exhibit. 2003.

[6] Lewitowicz, Jerzy, et al. "Modeling and Simulation of Flapping Wings Microaerial-vehicles Flight Dynamics." 26th International Congress of the Aeronautical Sciences (ICAS), Anchorage, AK. 2008.

[7] Cheng, B., and X. Deng. "Near-hover dynamics and attitude stabilization of an insect model." American Control Conference (ACC), 2010. IEEE, 2010.

[8] Ohishi, Kiyoshi, et al. "Microprocessor-controlled DC motor for load-insensitive position servo system." IEEE Transactions on Industrial Electronics 1 (1987): 44-49.

[9] Lehmann, Fritz-Olaf, and Simon Pick. "The aerodynamic benefit of wingwing interaction depends on stroke trajectory in flapping insect wings." Journal of experimental biology 210.8 (2007): 1362-1377.

[10] Dickinson, Michael H., Fritz-Olaf Lehmann, and Sanjay P. Sane. "Wing rotation and the aerodynamic basis of insect flight." Science 284.5422 (1999): 1954-1960.

[11] Sane, Sanjay P., and Michael H. Dickinson. "The aerodynamic effects of wing rotation and a revised quasi-steady model of flapping flight." Journal of experimental biology 205.8 (2002): 1087-1096.

[12] Cheng, Bo, et al. "Turning dynamics and passive damping in flapping flight." Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. IEEE, 2009.

[13] Nelson, Robert C. Flight stability and automatic control. Vol. 2. New York: WCB/McGraw Hill, 1998.

[14] Cheng, B., et al. "Aerodynamic damping during rapid flight maneuvers in the fruit fly Drosophila." Journal of Experimental Biology 213.4 (2010): 602-612.

[15] Finio, Benjamin M., and Robert J. Wood. "Open-loop roll, pitch and yaw torques for a robotic bee." Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.

[16] Kim, Joong-Kwan, and Jae-Hung Han. "Control effectiveness analysis of the hawkmoth Manduca sexta: a multibody dynamics approach." International Journal Aeronautical and Space Sciences 14.2 (2013): 152-161.

[17] Shim, Hyungbo, et al. "Yet another tutorial of disturbance observer: robust stabilization and recovery of nominal performance." Control Theory and Technology 14.3 (2016): 237-249.

[18] Li, Shihua, et al. Disturbance observer-based control: methods and applications. CRC press, 2016.

[19] Karsek, M., and A. Preumont. "Simulation of flight control of a hummingbird like robot near hover." Engineering Mechanics (2012): 322.

APPENDIX A: INERTIA COEFFICIENT

$$
\begin{aligned}
C_1 &= \frac{(I_y - I_z)I_z - I_{xz}^2}{I_x I_z - I_{xz}^2} \\
C_2 &= \frac{(I_x - I_y + I_z)I_{xz}}{I_x I_z - I_{xz}^2} \\
C_3 &= \frac{I_z}{I_x I_z - I_{xz}^2} \\
C_4 &= \frac{I_x z}{I_x I_z - I_{xz}^2} \\
C_5 &= \frac{I_z - I_x}{I_y} \\
C_6 &= \frac{I_{xz}}{I_y} \\
C_7 &= \frac{1}{I_y} \\
C_8 &= \frac{I_x(I_x - I_y) + I_{xz}^2}{I_x I_z - I_{xz}^2} \\
C_9 &= \frac{I_x}{I_x I_z - I_{xz}^2}
\end{aligned}
\tag{29}
$$

where I_x, I_y and I_z denotes moments of inertia with respect to the center of gravity location in the body fram.

Experimental study on the phase delay of low-cost IMU, low pass, and Kalman filter and its effect on the phase margin of angle estimation

S.L. Nam[1], S. Aurecianus[1], and H.V. Phan[2], T. Kang[1*], and H.C. Park[2]
[1]Department of Aerospace Information Engineering, Konkuk University, Seoul, South Korea
[2]Department of Smart Vehicle Engineering, Konkuk University, Seoul, South Korea

ABSTRACT

Nowadays, many low-cost MEMS inertial measurement units (IMUS) or motion processing units (MPUs) can be found in the market which can be used in control system. For the sensors to be useful in control system, the phase delay and frequency response should be given because it directly affects the stability of the closed loop system. However, almost all of the makers of low-cost MEMS IMUs or MPUs do not supply the phase characteristics. Furthermore, their performance characteristics under high frequency vibration are not given, even though they are very important when the sensors are used on drones or flapping MAVs.

In this paper, we developed an external measuring device that can accurately measure the rotational angle value from the vibrating MAVs. It is composed of 1-DOF gimbal with a servo motor to produce rotation and one rotary encoder to measure the rotation. The outer frames of the gimbals are made of carbon plates. At this time, the rotation angle is limited to +/- 90 degrees. The IMU to be evaluated is attached to the gimbal and it is connected to the microprocessor. Then, the servo command, encoder angle, and IMU data, which are obtained by the microprocessor, are sent to the computer for data processing and filter simulation. The filters used are low pass filter and Kalman filter. The experiments are conducted by rotating the IMU in sinusoidal and step pattern with and without flapping motion. Accordingly, the effect of filter parameters on estimated angle, including the phase delay, can be observed and studied.

1 INTRODUCTION

Over the years, low-cost MEMS IMUs and MPUs have been available on the market [1]. They are widely used in various applications, especially in small vehicles, such as micro aerial vehicle (MAV) or micro underwater vehicle (MUV). However, it is well known that the output values of such low-cost MEMS IMUs and MPUs contain a lot of noises. To solve this problem, the output value is filtered by software through low-pass filter, Kalman filter, complementary filter, or any other filters. However, if the filter is used, a phase delay occurs between the actual value and the estimated value. The phase delay due to the filters are added to the inherent delay of the sensor. The delay is an important factor in our analysis of flight data, or in controlling drones or MAVs. Also, via the gimbal system experiment, we can check the performance of the IMU under high frequency oscillation, which is critical if an IMU or MPU to be used in vibrating environment like flapping MAVs.

In this study, we developed a 1-DOF rotation measurement system consisting of a 1-DOF gimbal with one servo motor and two encoders. The servo motor is used to drive the rotating axis of the

*contact email: tskang@konkuk.ac.kr

gimbal and the encoders are attached to both sides of the shaft to measure the rotation angle. Later we found that one encoder is enough to get the reference angle information, and thus one encoder only is used to get the reference motion data in the experiment. Various designs of gimbal system can be found for different purposes such as controlling position of camera on MAV [2], attitude estimation and control [3], or removing bias [4]. Our design is different compared to the conventional ones in that it can mount directly the flapping MAVs and drive the gimbal with desired motion and we can get measured data from the IMU mounted on the flapping MAV as well as the reference motion data from the encoder so that we can evaluate the performance of the IMU mounted on the flapping MAV under more realistic flapping environment.

The flapping-wing MAV [5] containing the IMU is mounted on the gimbal system. The estimated angle values are obtained by calculating angle from accelerometer and applying low pass and Kalman filter using the rate information from the gyroscope in the IMU[6]. The estimated angle is compared with the reference angle data obtained from the encoder mounted on the shaft of the gimbal system.

2 Gimbal System

In this paper, we aim to develop a 1-DOF rotation measurement verification system to study the phase delay of low-cost IMU, low pass, and Kalman filter and its effect on the stability margin of feedback control system, as well as, to study the effect of vibration to the performance of small IMU

2.1 Mechanical design of the gimbal system

Figure 1 shows a 3D drawing of the target 2-DOF rotation measurement verification device to be fabricated for the future research. The currently developed 1-DOF rotation measurement verification device, which is shown in the Figure 2, consists of a gimbal, two encoders to measure the rotation angle, a supporter to support the gimbal and encoders, and a fixing device to fix the measurement object to the gimbal. Encoders were mounted on both sides of the gimbal so that the gimbals are not bent and they are being accurately measured. These encoders are used to measure the actual angle at which the gimbal is rotated.

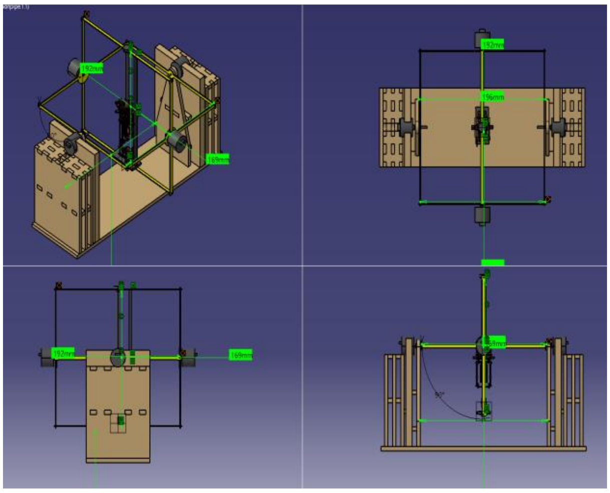


Figure 1 - Conceptual design of 2-DOF rotation measurement verification device for future research

The supporter and the fixing device were manufactured using a 5T-acrylic plate using a laser cutting machine. The gimbal was made of 1T-carbon plate using CNC.

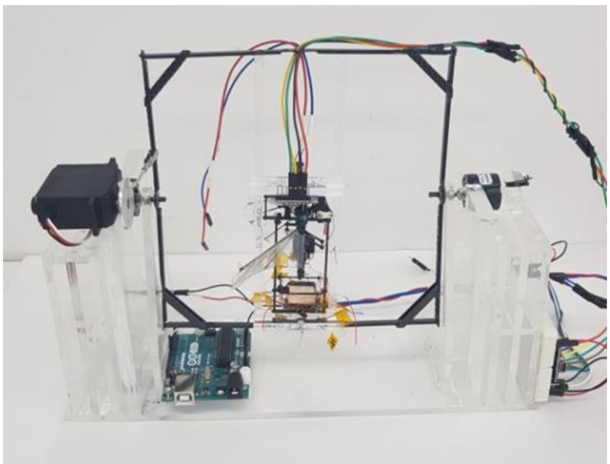


Figure 2 - 1-DOF rotation measurement verification device

2.2 Data acquisition of the 1-DOF gimbal system

Using the Mbed LPC1768, the encoder value was read through the SPI communication and transferred to the computer through another serial communication. The sampling frequency was 100Hz.

The encoder used for this device is AEAT-6012-A06, a high performance, low cost, optical absolute magnetic encoder module, designed to detect absolute reference angular position. The servo used for this device is BM-1301B. It is a digital servo made for robots. It was chosen because it has strong torque and accurate control over price.

2.3 operation test

Sine waves or square waves are generated in Arduino Uno board and the PWM signal corresponding to the motions are sent to the servo motor. A sine wave of 1Hz was generated for the operation test of the 1-DOF gimbal system. The amplitude was set as 20 degrees peak to peak.

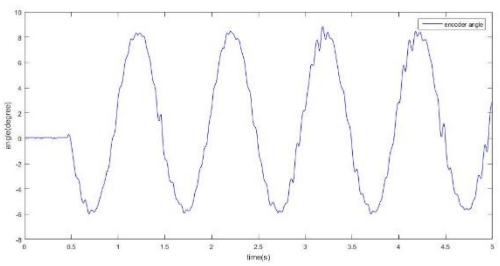


Figure 3 – result of operation test

The reference data obtained from the encoder are shown in Figure 3. The measured gimbal motion angle shows sine wave with 14 degrees peak to peak even though we applied sine wave with amplitude 20 degrees peak to peak. It is due to that the servo and the gimbal are connected through a mechanical linkage and the amplitude is scaled down due to the mechanical linkage. Thus all the data in this paper is scaled down due to the linkage effect.

3 Experiment

Currently, there are various types of inexpensive MPUs on the market. The IMU used in this experiment is MPU-9250. This sensor includes gyroscopes, accelerometer, and magnetometer. For convenience, we use sensor module and microprocessor MBED LPC1768.

3.1 Experiment configuration

Figure 4 shows a block diagram of the system for experiments. After the microprocessor MBED LPC1768 acquires data from sensor module, it processes the data and transfers it to the computer so that the data can be verified. The sampling rate of the sensors are set as 100 Hz.

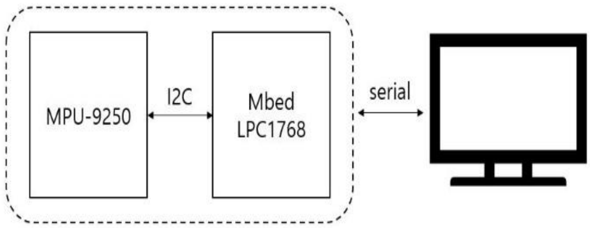


Figure 4 – IMU operation experiment block diagram

In this study, the data is filtered using a low-pass filter and a Kalman filter to obtain estimated angle. The experiment was conducted using KU-beetle of Konkuk University. We observe the phase delay phenomenon by comparing the encoder value coming out by rotating the gimbals and the value coming out through the filter by controlling the servo with Arduino Uno.

3.2 IMU data processing

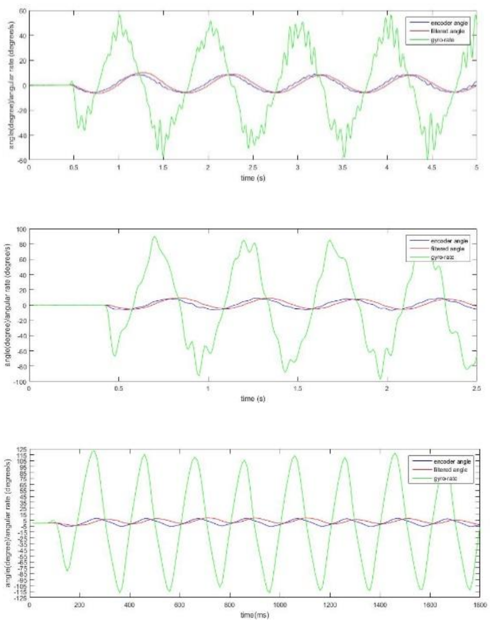
Among the data of IMU, only the gyro value of one axis and acceleration of three axes are used. The acceleration values of the three axes from the IMU are each passed through a low-pass filter. Then, the attitude angle is first extracted using the direction of the gravity acceleration measurement. The angle is calculated after a low-pass filter to move out the effect of vibration. The acceleration angle is calculated using equation:

$$\theta_{acc}[k] = \frac{360}{2\pi} tan^{-1}\left(\frac{a_x[k]}{\sqrt{a_y[k]^2 + a_z[k]^2}}\right) \quad (1)$$

The a_x, a_y, and a_z are measured and filtered acceleration signals for each axis. Then, the filtered angle was obtained by performing a Kalman filter processing on the gyro value of the single axis and the acceleration angle obtained above. The low pass and Kalman filter equation used in the simulation can be found in [4]

3.3 Experiment result

Experimental data were compared between encoder angle, filtered angle and gyro rate. In the first experiment, the amplitude of the signal sent to the servo was 20 degrees without flapping. There are four types of signals: a sine wave with periods of 1, 2, and 3 Hz and a periodic 0.5 Hz rectangular wave. The second test signal is a sine wave with an amplitude of 20 degrees and a period of 0.5 Hz. The difference from the first experiment is that FMAV flapping for a certain time.



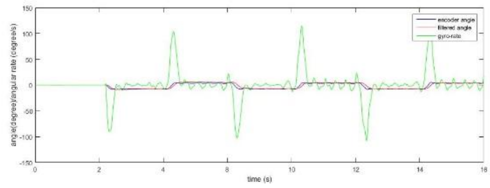


Figure 5 – non-flapping experiment

Figure 5 shows the result graph for the first experiment. From the top is a 1 Hz sine wave, a 2 Hz sine wave, a 5 Hz sine wave and a 0.5 Hz square wave. Comparing the filtered angle and the encoder angle, the phase delay of the filtered angle can be confirmed in all four graphs. This phase delay is critical especially if the closed loop is to have wide bandwidth. To overcome the phase delay of the sensing system, the closed loop controller should have enough phase margin greater than the phase delay of the sensing elment and the filtering software.

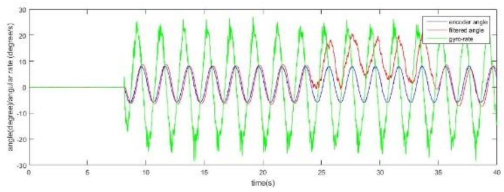


Figure 6 – flapping experiment

Figure 6 shows the result graph for the second experiment. The first 25 seconds were measured without flapping to see the obvious results. After then, 10 seconds of flapping is applied, and the flapping is stopped. It can be seen that the offset of the filtered angle increases by about 5 degrees during 10 seconds of flapping. The reason for this phenomenon is considered a result of a large vibration of the acceleration value during the flapping, and the accelerometer bias is affected by the vibration. Thus to improve the accuracy of the IMU, it is critical to use some mechanical vibration isolator to reduce the effect of mechanical vibration.

4 CONCLUSIONS

Through this study, the performance of low-cost IMU can be verified through a gimbal system, i.e., 1-DOF rotation measurement verification device. As can be seen from the experiment, the estimated value of the IMU through the filter has a phase delay phenomenon. The phase delay should be treated well to make a better closed-loop control system. Especially it is necessary for the closed loop control system to have enough phase margin to overcome the uncertainty due to the phase delay. Also the accelerometer is affected much by the high frequency vibration and it invokes offset change in the gyro-accelerometer filtering system. To reduce the error due to the offset change, it is necessary to use mechanical isolator to damp out high frequency oscillation

REFERENCES

[1] M. Perlmutter and L. Robin. High-performance, low cost inertial MEMS: A market in motion. In Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium, 2012.

[2] L. R. Manohar and C. M. Ananda. Design, simulation, and development of two axes gimbal for micro aerial vehicle. International journal of electrical, electronics and data communication. 03(05):62-65. 2015.

[3] F. Hoffmann, N. Goddemeier, and T. Bertram. Attitude estimation and control of a quadrotor. The 2010 IEEE/RSJ international conference on intelligent robots and systems. 2010.

[4] A. Roshanbin, H. Altartouri, M. Karasek, and A. Preumont. COLIBRI: a hovering flapping twin-wing robot. International journal of micro air vehicles. 9(4):270-282. 2017.

[5] H.V. Phan and H.C. Park. Remotely controlled flight of an insect-like tailless flapping-wing micro air vehicle. The 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 2015.

[6] G.M. Siouris. An engineering approach to optimal control and estimation theory, John Wiley & Sons, Inc, New York, NY, 1996.

Attitude Control of Tiltwing Aircraft Using a Wing-Fixed Coordinate System and Incremental Nonlinear Dynamic Inversion

F. Binz,* T. Islam and D. Moormann
RWTH Aachen University, Templergraben 55, 52062 Aachen, Germany

Abstract

In this paper we present a novel concept for robustly controlling the attitude of tiltwing aircraft. Our main contribution is the introduction of a wing-fixed coordinate system for angular acceleration control, which forms the basis of a simple and robust attitude controller. Using the wing-fixed coordinate system allows us to describe the actuator effectivity using simple approximations based on the current operating conditions of the aircraft. Coupled with a robust angular rate control concept, which does not rely on an accurate aerodynamic model, we present a controller stabilizing the entire flight envelope of a tiltwing aircraft. The underlying attitude acceleration controller uses the concept of Incremental Nonlinear Dynamic Inversion (INDI) to achieve robustness against aerodynamic uncertainties. The resulting controller is evaluated in both simulation studies and flight tests.

1 Introduction

A key goal in the design of many unconventional aircraft types is the combination of efficient forward flight with vertical take-off and landing (VTOL) capabilities. One solution to this problem is the concept of a tiltwing aircraft. These aircraft fly like a conventional airplane in forward flight and achieve VTOL capabilitites by tilting the entire wing upwards to hover.

To stabilize the aircraft in both hover and forward flight several actuators are needed. The aircraft considered here is depticted in Figure 1 and features the following actuators for attitude control: asymmetric thrust of the main motors, ailerons, elevator and thrust of the auxiliary motor. Table 1 shows the primary moments induced by each actuator during hover and forward flight and the corresponding tilt angle. Table 1 hints at a central problem in the design of attitude controllers of tiltwing aircraft: the moments induced by the asymmetric thrust and ailerons change direction between hover and forward flight. Consider the ailerons as an example: During hover flight, at a tilt angle of 90°, the ailerons primarily induce a yawing moment, since they are positioned in the slip stream of the main engines. However, during forward flight, at a tilt angle of 0°, the ailerons primarily incude a rolling moment, as in a conventional airplane. In between the hover and forward flight configurations the ailerons incude both a rolling and a yawing moment. Besides the change in direction of the actuator-induced moments, the transition between hover and forward flight is further characterized by potentially highly turbulent airflow behind the main wing. This complicates the design of high-fidelity aerodynamic models, which are needed for many advanced control schemes. Because of these properties, controller design for tiltwing aircraft still presents a challenging problem.

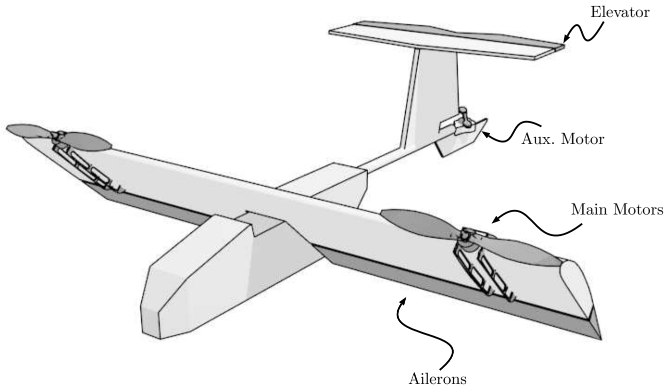


Figure 1: Example tiltwing aircraft in hover configuration

	hover flight	**forward flight**
asym. throttle δ_{asym}	roll	yaw
aux. throttle δ_{aux}	pitch	(-)
ailerons ξ	yaw	roll
elevator η	(-)	pitch
tilt angle σ	90°	0°

Table 1: Actuator effectivity in hover and tilt configuration

In light of these properties, our main contributions are:

- We observe that the moments induced by the actuators only change direction w.r.t. the body-fixed coordinate system. In the wing-fixed coordinate system the direction of the induced moments is constant. Of course, this simple – and in hindsight obvious – observation by itself does not lead to more robust controllers, but should be

*binz@fsd.rwth-aaachen.de

understood as a tool for understanding tiltwing aircraft dynamics and exploring the design space of attitude controllers.

- Inspired by the body of work concerning robust control schemes in recent years (e. g. Incremental Nonlinear Dynamic Inversion (INDI) [6] or Incremental Backstepping [1]), we propose the combination of a wing-fixed coordinate system with an attitude acceleration controller based on the principle of INDI to yield a robust attitude controller. In the spirit of INDI, which is a sensor based control concept, we derive the needed actuator effectivity not on the basis of characteristic maps at certain trimmed flight states, but instead use the available measurements and simple empirical and analytical models to estimate the actuator effectivity in the current operating conditions.

2 The Wing-Fixed Coordinate System

In this paper we are exclusively considering tiltwing aircraft with a single tiltable wing as depicted in Figure 1 and described in [4]. Our approach should however be also applicable to other tiltwing aircraft with only slight modifications (i.e. quad-tiltwing designs [2]).

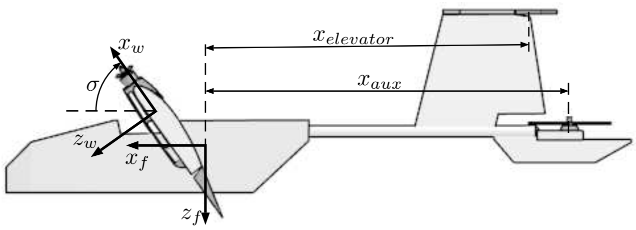


Figure 2: Side view of the aircraft

Figure 2 shows the wing-fixed coordinate system. Conceptually, the origin of the wing-fixed coordinate system lies in the tilt axis of the wing. Since the center of gravity and tilting axis are small, we don't consider the distance between the origins of the body-fixed and wing-fixed coordinate systems in the following treatment.

To transform a vector given in the body-fixed coordinate system (Index b) to the wing-fixed coordinate system, we define the following transformation matrix:

$$T_{wb} = \begin{bmatrix} \cos\sigma & 0 & -\sin\sigma \\ 0 & 1 & 0 \\ \sin\sigma & 0 & \cos\sigma \end{bmatrix} \tag{1}$$

The tilt angle σ assumes a value of ca. $90°$ resp. $0°$ in hover resp. forward flight configuration.

2.1 Transformation of Moment of Inertia

For the scope of this work, we assume that the aircrafts moment of inertia is constant w.r.t. the tilt angle σ. Nevertheless, we need to transform the moment of inertia given in the body-fixed coordinate system into the wing-fixed coordinate system. The moments acting on the aircraft expressed in the body-fixed coordinate system M_b are linked to the body-fixed accelerations $\dot{\Omega}_b$ by the body-fixed moment of inertia J_b:

$$M_b = J_b \cdot \dot{\Omega}_b \tag{2}$$

The body-fixed moments M_b can be expressed in the wing-fixed coordinate system using the transformation matrix T_{wb}:

$$\begin{aligned} M_w &= T_{wb} \cdot M_b & (3) \\ &= T_{wb} \cdot J_b \cdot T_{wb}^{-1} \cdot \dot{\Omega}_w & (4) \\ \Rightarrow J_w &= T_{wb} \cdot J_b \cdot T_{wb}^{-1} & (5) \end{aligned}$$

Using (5) we calculate the wing-fixed inertia based on the current tilt angle and the body-fixed inertia.

2.2 Actuator Effectivity

We introduced the wing-fixed coordinate system with the main goal of simplifying the description of the actuator effectivity. For attitude control, we are interested in the actuator effectivity concerning the roll, pitch and yaw moments (L, M and N). In this section we will discuss the models we employ for the different actuators available. When modelling the actuator effectivity, we try to find simple models which still result in satisfactory controller performance. We are thus neglecting various effects, the most important of which are:

- No cross-coupling between actuators. Every actuator only induces a moment along one axis in the wing-fixed coordinate system.
- Every actuator is exposed to the same free stream velocity, disregarding effects like downwash from the main wing onto the elevator.

2.2.1 Motor model

We assume that the thrust produced by a fixed-pitch propeller is primarily influced by two factors: the angular velocity of the propeller and the inflow speed. Based on this assumption, we first introduce a thrust model

$$F_{motor} = f(V, \delta) \tag{6}$$

where δ is the throttle signal corresponding to the motor currently considered. Based on prior measurements of electric motors performance, we assume that the angular velocity of a an electric motor is approximately linear to the throttle signal applied to the electronic speed controller. This means, that – lacking a direct measurement of the propeller angular velocity – the throttle signal can be used as an equivalent signal.

The thrust model $F_{motor}(V, \delta)$ can be obtained in different ways. In our work, we obtained the thrust model using a simulation based on semi-analytical formulas supported by empirical static-thrust data. The resulting model is a two-dimensional characteristic map, which we approximate using a two-dimensional second-order polynomial. The local derivative of this polynomial is then computed online.

We use this motor model for the asymmetric thrust produced by the main motors and the thrust produced by the auxiliary motor. For the auxiliary motor, we make the simplifying assumption, that the inflow is negligible ($V_{aux} = 0$), thus reducing the model complexity further.

The inflow of the main motors is estimated using the currently measured airspeed V_A, transformed into the wing-fixed coordinate system:

$$\vec{V}_{Aw} = \begin{bmatrix} u_{Aw} \\ 0 \\ w_{Aw} \end{bmatrix} = T_{wb} \cdot \begin{bmatrix} V_A \\ 0 \\ 0 \end{bmatrix} \tag{7}$$

Only the x-component u_{Aw} of $\vec{V}_{Aw}$ is used as an input to the thrust model.

Using the lever arms of the main motors, we obtain the following effectivity of the main motors:

$$\frac{\partial N}{\partial \delta_{asym}} = 2 \cdot y_{motor} \cdot \frac{\partial F_{motor}(V, \delta)}{\partial \delta_{asym}}\Big|_{V=u_{Aw}, \delta=\delta_{sym,0}} \tag{8}$$

Here, $\delta_{sym,0}$ denotes symmetric throttle signal in the current controller timestep and y_{motor} denotes the lever arm between the motor and the aircrafts center of mass. The factor 2 accounts for the two motors, one on each side of the aircraft.

Similarly, for the auxiliary motor we obtain:

$$\frac{\partial M}{\partial \delta_{aux}} = x_{aux} \cdot \frac{\partial F_{aux}(V, \delta)}{\partial \delta_{aux}}\Big|_{V=0, \delta=\delta_{aux,0}} \tag{9}$$

2.2.2 Control surfaces

We distinguish between two different kinds of control surfaces: those which are assumed to be completely in the free stream and those which are in the slip stream of a propeller. Both kinds of control surfaces are modeled as thin plates of finite length, where the lift F_{lift} changes with the control surface deflection δ according to the following equation:

$$\frac{\partial F_{lift}}{\partial \delta} = \frac{\rho}{2} V^2 S \cdot 2\pi \cdot \frac{\Lambda}{\Lambda + 2} \tag{10}$$

Here, ρ is the air density, V the inflow speed, S the control surface area and Λ the aspect ratio of the wing corresponding to the control surface.

2.2.3 Elevator

Based on (10) the elevator effectivity with lever arm $x_{elevator}$ is then given by

$$\frac{\partial M}{\partial \eta} = \frac{\rho}{2} V^2 S \cdot 2\pi \cdot \frac{\Lambda}{\Lambda + 2} \cdot x_{elevator} \tag{11}$$

The velocity V is assumed to be equal to the measured aerodynamic speed V_A.

2.2.4 Ailerons

A characteristic feature of the tiltwing configuration we describe here is, that the ailerons are partly within the slip stream of the main motors. To capture this property, the wing is divided into three sections, see Figure 3.

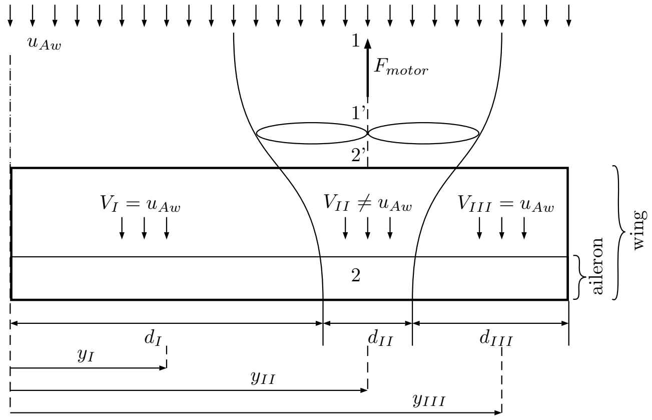


Figure 3: Division of the wing into three sections, with different inflow speeds

Sections I and III are in the free stream and are modeled similar to the elevator, where the free stream speed is used as the inflow speed. Section II is completely in the slip stream of the propeller. Here, the propeller induces an inflow speed even if there is no free stream speed. To estimate this inflow speed, we apply the Bernoulli-Equations along a streamline to the two control volumes $1 \rightarrow 1'$ and $2' \rightarrow 2$:

$$p_{1'} - p_{2'} = \frac{\rho}{2}(V_1^2 - V_2^2) \tag{12}$$

Multiplying (12) by the propeller area S_{disk} gives the thrust F_{motor} produced by the motor. We assume that the inflow speed equals the free stream speed ($V_1 = u_{Aw}$) and the slip stream is fully developed ($V_2 = V_{II}$).

$$F_{motor} = S_{disk} \frac{\rho}{2}(u_{Aw}^2 - V_{II}^2) \tag{13}$$

This results in the following expression for the inflow speed V_{II} of section II:

$$V_{II} = \sqrt{\frac{F_{motor}(\delta, u_{Aw})}{S_{disk}} \frac{2}{\rho} + u_{Aw}^2} \tag{14}$$

Here, F_{motor} is the thrust obtained by evaluating the thrust model (6) at the current operating point. Using this inflow speed, the effectivity of the ailerons in the slip stream of the motors can be calculated. In total, the effectivity of the ailerons is the sum of the effectivity of the individual sections, resulting in:

$$\frac{\partial M}{\partial \xi} = 2 \cdot \frac{\rho}{2} \cdot 2\pi \frac{\Lambda}{\Lambda + 2} \sum_{i=I..III} V_i^2 \cdot S_i \cdot y_i \tag{15}$$

Again, the factor 2 accounts for the two ailerons.

Summarizing, the moments induced by the actuators are given by:

$$\underbrace{\begin{bmatrix} \partial L \\ \partial M \\ \partial N \end{bmatrix}}_{\partial \vec{M}} = \underbrace{\begin{bmatrix} (15) & 0 & 0 & 0 \\ 0 & (11) & (9) & 0 \\ 0 & 0 & 0 & (8) \end{bmatrix}}_{M_\delta(u_{Aw}, \delta_{sym}, 0)} \cdot \underbrace{\begin{bmatrix} \partial \xi \\ \partial \eta \\ \partial \delta_{asym} \\ \partial \delta_{aux} \end{bmatrix}}_{\vec{\delta}} \tag{16}$$

3 Attitude Control using Incremental Nonlinear Dynamic Inversion (INDI)

The actuator effectivity described in section 2.2 is needed to apply the concept of INDI to our aircraft. The theory underlying INDI is presented in [6] and [7] and is not repeated here. [6] and [7] are slightly different formulations of INDI, we employ the formulation described in [7]. The central underlying assumption is that the so called *time scale separation principle* holds w.r.t. the actuator dynamics and the dynamics of aerodynamic forces and moments. The control signal can then be computed incrementally using the actuator effectivity matrix M_δ given in (16) using the following formula:

$$\vec{\delta}_{k+1} = \vec{\delta}_k + M_\delta^+ \cdot J_w \cdot (\vec{\nu} - \dot{\vec{\Omega}}) \tag{17}$$

Here, M_δ^+ is the pseudo-inverse of M_δ, which is not invertible due to its dimension and is rank-deficient in hover mode since the elevator effectivity is zero. $\vec{\nu}$ is the pseudo-control input for the INDI control loop and is a vector of commanded angular accelerations. The prospect of INDI as described in [7] is, that closed-loop dynamics from ν_i to $\dot{\Omega}_i$ equals the corresponding actuator dynamics, where i corresponds to the moment axis ($i = x, y, z$). That is, the roll acceleration loop has the dynamics of the aileron actuators. The actuator dynamics are modeled for each actuator individually as first-order lags and are denoted as $A_i(z)$. We note that the pitch axis is actually over-actuated during the transition and fast-forward flight, since both the elevator and the auxiliary motor thrust are effective. For the purposes of designing the attitude controller, we use the slower dynamics of the auxiliary motor to derive attitude controller gains. Figure 4 shows the structure of the

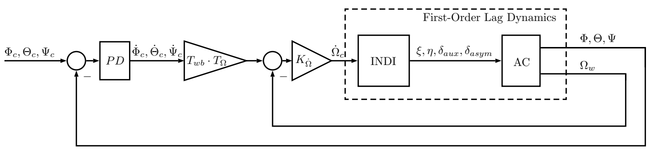


Figure 4: Attitude controller loop

attitude controller. To close the attitude control loop, we use a traditional cascaded approach, where both the angular rates and the attitude angles are fed back using proportional ($K_{\dot{\omega}}$) resp. proportional-derivative (PD) controllers. To generate the wing-fixed commanded angular rates from attitude angle errors, the commanded attitude rates $\dot{\Phi}_c, \dot{\Theta}_c, \dot{\Psi}_c$ are first multiplied by the inverted attitude angle dynamics T_Ω (18) and then transformed into the wing-fixed coordinate system using the transformation matrix T_{wb} (1).

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\Theta \\ 0 & \cos\Phi & \sin\Phi\cos\Theta \\ 0 & -\sin\Phi & \cos\Phi\cos\Theta \end{bmatrix} \cdot \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} \tag{18}$$

4 Simulation and Experimental Results

To validate the controller design presented in Sections 2 and 3 we first conducted extensive simulation studies. We analyze the perfomance of the attitude controller over the entire flight envelope in Section 4.1. In Section 4.2 we discuss results obtained from flight tests.

4.1 Simulation studies

To conduct the simulation experiments we used the simulation environment described in [5]. The aircraft is modelled as a set of components, where aerodynamic interaction between certain components is simulated to capture the characteristics of tiltwing aircraft.

4.1.1 Stability over entire flight envelope

For tiltwing aircraft, stability during the transition between hover and forward flight is of course of utmost importance. To ensure, that the aircraft is stabilized even when the tilt angle changes quickly, we conducted simulation experiments where the tilt angle is reduced linearly from its hover configuration ($\sigma = 90°$) to its forward flight configuration ($\sigma = 0°$) at different tilt speeds $\dot{\sigma}$. Figure 5 shows the resulting error in the pitch angle over time, normalized to the time when the tilt angle reaches $0°$. Overall, the pitch angle is stabilized very well over the entire flight envelope. The tilt speed $\dot{\sigma}$ intially has a minor effect on the pitch angle error, where faster changes result in higher pitching moments and thus larger pitch angle errors. Still, the maximum pitch angle is below $1°$, which would be a significant improvement compared to previous controller designs for this aircraft.

4.1.2 Attitude controller performance and robustness

Figure 6 shows the step response of the attitude controller in hover configuration, where the computed actuator effectivity is multiplied by different gains to simulate modeling errors. Some cross-coupling between the body-fixed roll and yaw axes exists, mainly because the inverted dynamics (18) does not take the different actuator dynamics into account. In the nominal case $(\times 1.0)$ the controller shows satisfactory performance with little oscillations or overshoot. To test the robustness of the attitude controller, we multiplied the calculated actuator effectivity with factors of 0.5 or 1.5 respectively. Despite these large errors in the actuator effectivity, the attitude controller was still stable. Similar to results found in [8],

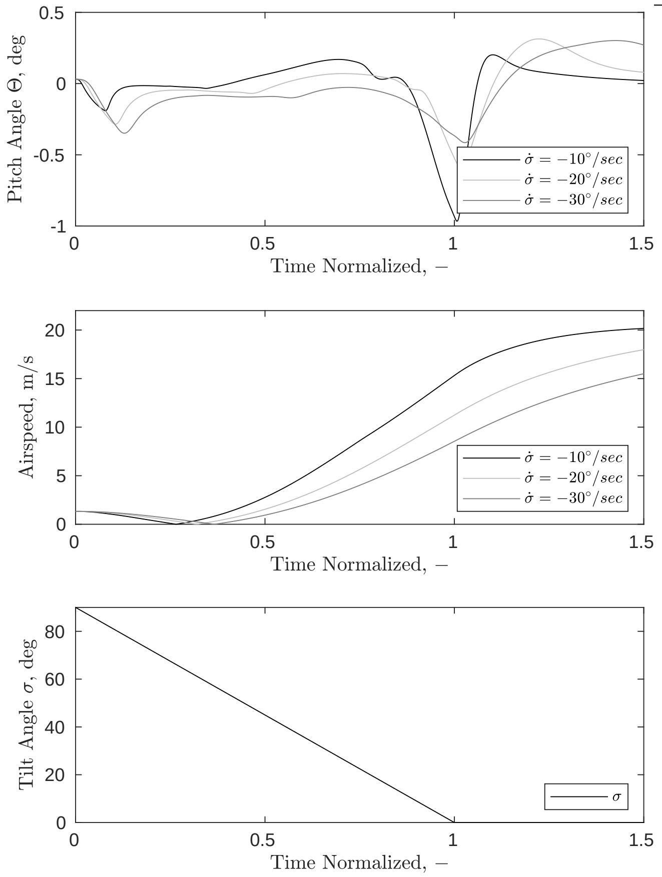


Figure 5: Pitch angle error during transition at different transition speeds (simulation)

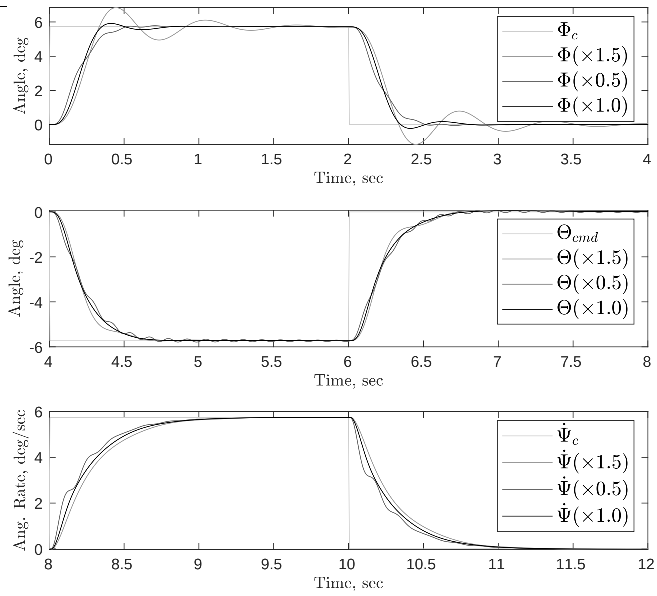


Figure 6: Attitude controller performance in hover flight (simulation)

assuming a too large actuator effectivity results in a somewhat more sluggish response, while assuming a too small actuator effectivity leeds to fast oscillations.

We conducted similar simulation studies for fast forward flight, see Figure 7. Here, the yaw rate $\dot{\Psi}$ is commanded as $\dot{\Psi}_c = \frac{g}{V_A} \tan \Phi_c$, which is the yaw rate needed for turning without side-slip. Overall, the step responses are still acceptable, but show some overshoot and more sluggish behaviour. We found, that the main cause for this difference is the innermost INDI control loop. Here, the expected accelerations differ significantly from the actual accelerations in fast forward flight. This seems to be a result of quite large damping moments which arise in fast forward flight. In the derivation of INDI these terms are assumed to be negligible. We are currently investigating this issue further to gain deeper understanding and find possible mitigations.

Further robustness analyses, for example against time delays, vibrations, changing actuator dynamics and nonlinearites like saturation in actuator dynamics, were investigated in simulation studies. The results match those already reported in the literature [6, 7, 8] and are thus not repeated here.

Summarizing, we conclude that the presented controller should be able to robustly stabilize tiltwing aircraft in their entire flight envelope.

4.2 *Flight Tests*

We conducted the first flight tests in hover flight mode. To assess the performance of the innermost INDI angular acceleration loop, we compared the expected angular accelerations to the measured angular accelerations. The expected angu-

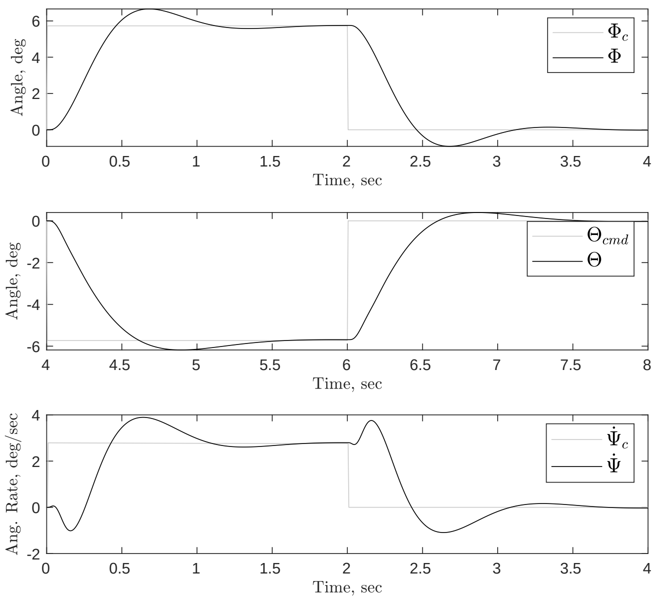


Figure 7: Attitude controller performance in fast forward flight (simulation)

lar accelerations were obtained using the theoretical model of the closed-loop angular acceleration dynamics $A_i(z)$ (see section 3). Figure 8 shows an exemplary timeseries of the wing-fixed yaw and pitch accelerations, $\dot{r}$ resp. $\dot{q}$. It is evident,

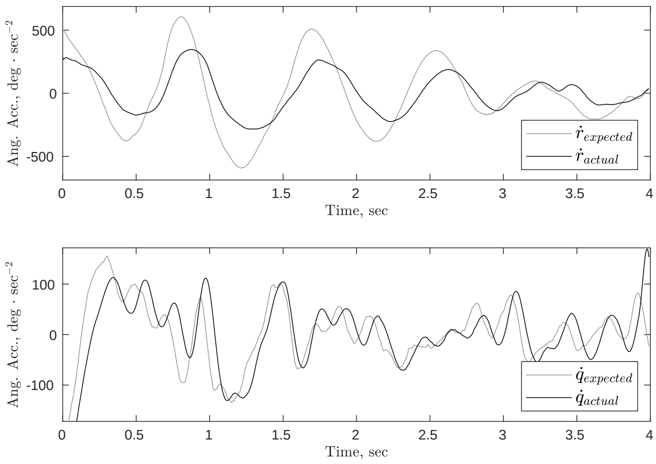


Figure 8: Comparison of expected and actual angular accelerations in hover flight (experiment)

that the wing-fixed yaw accelerations differ significantly from the expected yaw accelerations. We expect that the main cause of this is an inaccurate motor model, which currently does not take the battery voltage into account. Sampling of the characteristic map for the main motor thrust thus results in an overestimation of the effectivity of the asymmetric thrust. Because of this overestimation, the expected acceleration due to asymmetric thrust is higher than the actual acceleration.

The pitch acceleration does not show this particular behaviour, because instead of varying the auxiliary motor speed, the auxiliary motor was operated in a fixed-speed mode with a variable pitch propeller. Thus we expect that the yaw acceleration will improve significantly once the battery voltage taken into account in the thrust model.

The wing-fixed roll accelerations are not shown here, because the effectivity of the ailerons is very low in hover flight. The roll accelerations are thus dominated by the highly turbulent airflow and wind gusts.

Figure 9 shows the performance of the attitude controller described in section 3. Both, the body-fixed roll and pitch axes are stabilized by the controller. Despite the difference in expected and actual wing-fixed yaw acceleration, the performance of the roll controller is satisfactory.

After successful flight tests in hover mode, we conducted further flight tests to validate the controller in the entire flight envelope. Similar to the simulation study summarized in Figure 5, we conducted a transition from hover flight to fast forward flight. Figure 10 shows the roll and pitch angle controller performance during the entire transition phase. The roll and pitch angles were directly commanded by the pilot. The absolute error in roll and pitch angle stays below $3°$ resp. $1°$

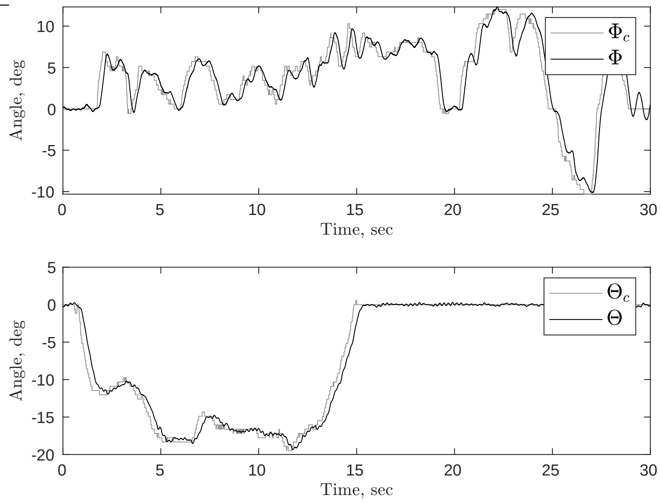


Figure 9: Attitude controller performance in hover flight (experiment)

during the entire transition phase, thus showing a very good correspondence to the simulation studies presented earlier. There are minor oscillations in both pitch and yaw, which are incudced by the controller. Since the controller used to generate these results was not tuned in free flight tests prior these experiments, we are confident that a less agressive parameter selection in the attitude controller will mitigate the oscillations.

We (unintentionally) observed the good robustness properties of the INDI controller during the transition from fast forward flight to hover flight. Figure 11 shows the pilots attempt to transition back into hover flight by increasing the commanded tilt angle from $0°$ to about $45°$ during the first 25 seconds. However, because of the high airspeed of about $19\,\mathrm{m\,s^{-1}}$, the servo responsible for tilting the wing was not able to rotate the wing against the aerodynamic forces. Since there is currently no servo position feedback, the INDI controller assumes that the commanded tilt angle σ_c equals the actual tilt angle (estimated to σ_{est}). Despite the large difference between commanded and actual tilt angle, the INDI still was able to stabilize the aircraft without any noticable perfomance degradation. When further increasing the commanded tilt angle σ_c to about $60°$ from seconds 25 to 30, the difference between the expected and actual tilt angle evidently became too large, leading to instability.

Starting at second 35, the pilot began to reduce the airspeed by first reducing the commanded symmetric throttle. At an airspeed of about $17\,\mathrm{m\,s^{-1}}$, the tilt servo was able to overcome the aerodynamic forces, resulting in the wing abruptly tilting up. This abrupt motion of the entire wing of course induces large disturbances, leading to errors of $40°$ and $10°$ in the roll resp. pitch axes. The controller was however able to stabilize the aircraft quickly in about $1\,\mathrm{s}$. In terms of controller performance, we think this incident examplifies the

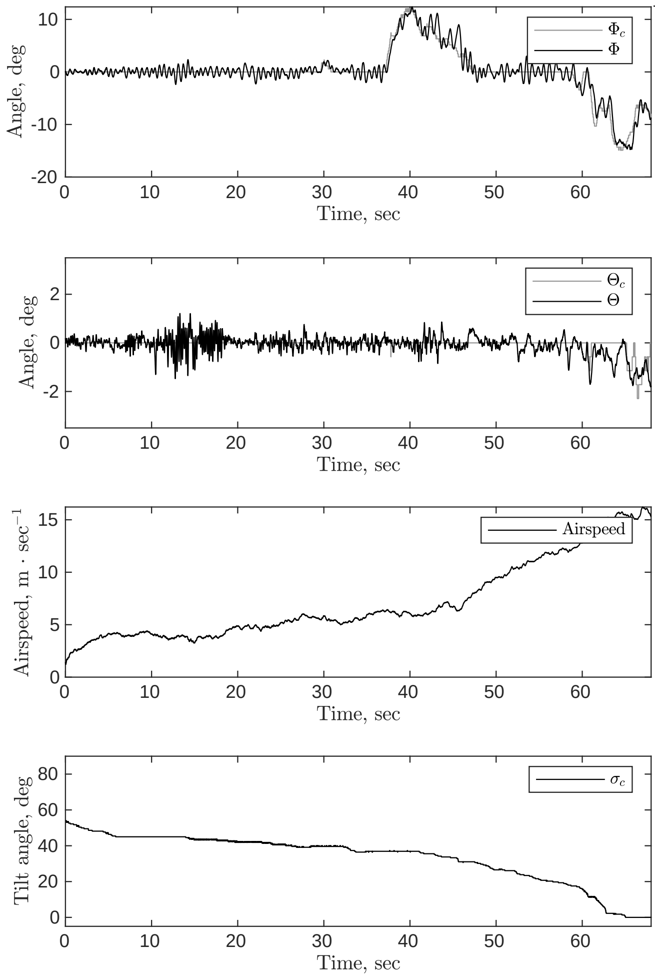


Figure 10: Attitude controller during transition from hover to forward flight (experiment)

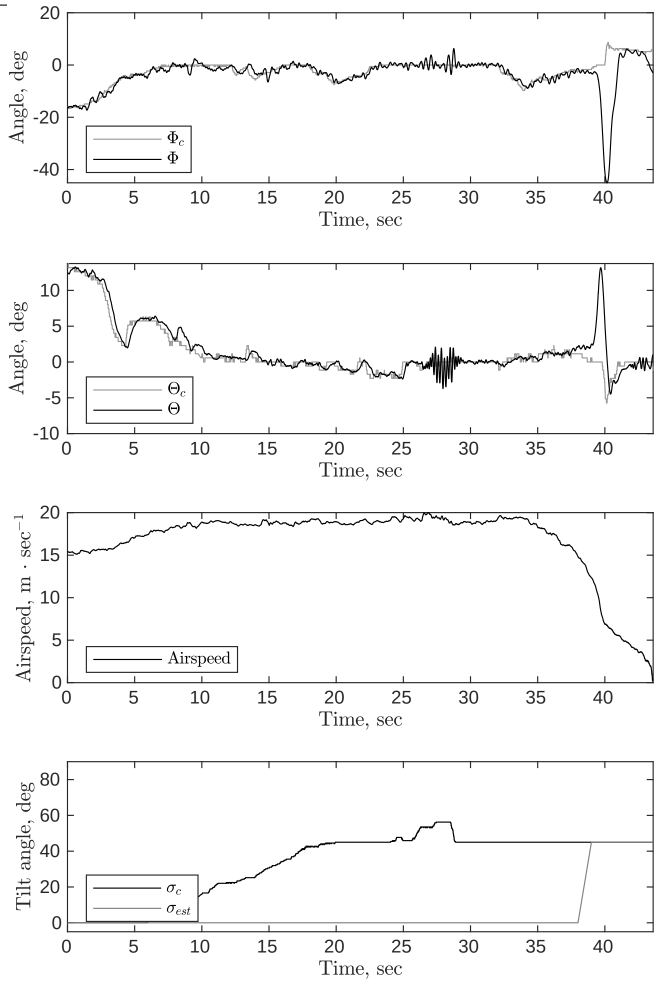


Figure 11: Attitude controller during transition from forward flight to hover (experiment)

good disturbance rejection qualities of the INDI controller.

5 Related Work

The literature offers several alternative approaches for the attitude control of hybrid VTOL aircraft. Naturally, the main differences between the different approaches concern the way in which the highly variable dynamics between hover and forward flight are handled.

In [4] and [2] the aircraft dynamics are linearized around certain trimmed flight states at different airspeeds. This is either done using aerodynamic models of the aircraft or using wind-tunnel measurements. Based on the linearized aircraft dynamics in a flight state a suitable controller can be found. The main assumption made by the authors is that the airspeed is quasi-stationary w.r.t. to the attutide dynamics. Thus, the airspeed can be used as a gain-scheduling variable for the attitude controller. This approach was demomstrated to work well, as long as the underlying assumption of a slowly varying airspeed is met. However, a signficant performance degradation was observed in the aircraft described in [4] once the airspeed varied non-stationary. Also, both approaches are heavily dependent on an accurate aerodynamic model. When obtaining accurate aerodynamic models using wind-tunnel experiments, this results in quite a substantial effort needed once the aircraft is changed (e.g. new wing or fuselage design).

A similar approach to the one we present in this work is found in [3]. Here the authors employ a form of INDI for the attitude control of a tiltrotor aircraft based on the currently measured fight state. However, instead of only considering the actuator effectivity, the incremental change in actuator input $d\delta$ is based on an full aerodynamic model of the aircraft. The authors thus arguably loose some of the benefits of INDI as described in [6, 7] (i. e. robustness against aerodynamic model uncertainty) in favor of a potentially better controller performance. On top of the attitude control itself, the authors also include an actuator allocation procedure to deal with saturations and varying actuator effectivity. Using simulation studies, this approach was reported to show good results over the entire flight envelope.

6 Conclusion and Future Work

In his work, we presented the application of INDI to a tiltwing aircraft. Using a wing-fixed coordinate system, we derived simple formulas to describe the actuator effectivity with the objective to minimize the dependency on wind-tunnel measurements or high-fidelity aerodynamic models. The resulting formulas only depend on geometric properties of the aircraft and characteristic maps of the thrust produced by the motors, which can be quite easily measured. Thus, this new attitude controller concept should be easily adaptable to changing aircraft designs. The resulting attitude controller showed good performance and robustness properties in simulation studies.

We conducted free flight tests in hover flight mode and obtained similar behaviour to the simulation results. Subsequent flight tests then covered the entire flight envelope from hover flight to fast forward flight. The controller showed good performance in conjunction with good robustness and disturbance rejection qualities.

Future work will concentrate on practical issues like disabling the auxiliary motor during fast forward flight or dealing with actuator saturations in a principled manner.

References

[1] Paul Acquatella B., Erik-Jan Van Kampen, and Q P Chu. Incremental Backstepping for Robust Nonlinear Flight Control. In *Proceedings of the EuroGNC 2013, 2nd CEAS Specialist Conference*, 2013.

[2] Y Beyer, T Krüger, A Krüger, M Steen, and P Hecker. Simulation and Control of a Tandem Tiltwing RPAS Without Experimental Data. In *International Micro Air Vehicle Conference and Flight Competition (IMAV)*, pages 162–169, 2017.

[3] Gabriele Di Francesco and Massimiliano Mattei. Modeling and Incremental Nonlinear Dynamic Inversion Control of a Novel Unmanned Tiltrotor. *Journal of Aircraft*, 53(1):73–86, 2016.

[4] Philipp Hartmann, Carsten Meyer, and Dieter Moormann. Unified Velocity Control and Flight State Transition of Unmanned Tilt-Wing Aircraft. *Journal of Guidance, Control, and Dynamics*, 40(6):1348–1359, 2017.

[5] Marten Schütt, Tobias Islam, Philipp Hartmann, and Dieter Moormann. Scalable Design Approach to Analyse Flight Mechanical Performance for Tiltwing UAVs. In *International Council of Aeronautical Sciences, ICAS, 2018-09-09 - 2018-09-14, Belo Horizonte, Brazil*, 2018.

[6] S Sieberling, Q P Chu, and J A Mulder. Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction. *Journal of Guidance, Control, and Dynamics*, 33(6):1732–1742, 2010.

[7] Ewoud J J Smeur, Qiping Chu, and Guido C H E de Croon. Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles. *Journal of Guidance, Control, and Dynamics*, 39(3):450–461, 2016.

[8] R C van't Veld, E van Kampen, and Q P Chu. Stability and Robustness Analysis and Improvements for Incremental Nonlinear Dynamic Inversion Control. (January), 2018.

Towards Micro Aerial Manipulation Using a Computational Compensation Strategy

Aaron Eleazar Lopez Luna *[1], Jose Martinez-Carranza[1,2], and Israel Cruz-Vega[1]

[1]Instituto Nacional de Astrofisica, Optica y Electronica, Mexico
[2]University of Bristol, UK

ABSTRACT

This paper presents an aerial manipulation system for low-cost micro air vehicles, which consist of two degrees of freedom robotic arm attached to the lower part of the vehicle, this vehicle is a commercial aerial vehicle parrot bebop-2. Thus, we propose to extend the capabilities of this inexpensive vehicle towards aerial manipulation. For the latter, this work presents a novel structure design which allows the bebop-2 to carry a manipulator in the lower part of the structure. The conventional proportional-integral-differential (PID) control algorithm used in most of these kind systems is not sufficient to deal with the new stability problems involved in this novel system. Therefore, to improve the control effectiveness, a computational-based compensation strategy based on the k-nearest neighbors (KNN) algorithm is incorporated into the control loop. KNN strategy can provide the adequate compensation at a low computation cost and is promising for real-world application. Experimental results developed in this work demonstrate a satisfactory performance for the proposed robotic arm design and the proposed control technique.

1 INTRODUCTION

Mobile manipulators (robotic manipulator arms attached to mobile bases) research has grown in recent years due to the importance and popularity of this useful systems in industrial and commercial applications. Ground mobile manipulators have been researched for use in areas like marine, agriculture, space and industrial applications [1, 2, 3]. Although many works and research mainly focus on the use of mobile ground systems, aerial vehicles have become widespread and more pervasive in research and technological development. Aerial vehicles are attractive due to their navigation capability in large areas where humans can not access or where mobile ground systems [4] may not perform adequately. In addition to the plethora of applications that can be developed with aerial systems (e.g., precision agriculture, video photography, infrastructure inspection, etc.), there is a growing interest for combining these aerial platforms with robotic manipulators, by attaching them to the aerial structure in order to obtain a new configuration of aerial manipulation systems [5]. Aerial systems like Unmanned Aerial Vehicles (UAVs) or drones, can be either controlled from the ground station, or by autonomous on board control algorithms. The interest in using this type of system comes not only from its dynamics, which represent an attractive control problem but also from the design issue. Notably, the researchers focus on the optimization of operational algorithms [6, 7, 8].

According to literature, the quadcopter is one of the most efficient configurations to implement an aerial manipulator due to their superior mobility in comparison with other available configurations [1, 2, 8, 6, 7]. Aerial manipulators open a new application area for robotics and aerial systems. Nevertheless, this new configuration represents a new problem in the stability control of the aerial vehicle. Movements of a manipulator attached to a VTOL during flight mode bring about disturbances which can cause instability and the loss of the entire system. New models and control algorithms have been proposed to prevent this situation [4, 9]. The leading proposals to ensure an admissible flight performance are; restricting the movement of the manipulator, incrementing the torque of actuators of the system, consider the change of mass distribution in the model and consider the influence of the motion of manipulator in system dynamics [10].

In this work, we consider the proposal of taking into account the changing of the behavior of the UAV due to the movement of the arm and we propose an experimental study to determine the variation in plant dynamics and an appropriate correction in the attitude control to approximate the real trajectory of the system to the desired trajectory. Therefore in this work we propose a novel design of an aerial manipulator arm with two degrees of freedom (DOF) attached to a commercial quadcopter parrot bebop-2. We also propose to incorporate a computational compensation strategy to a Classical PID control to ensure the stability of the proposed aerial manipulator. We determine the values for the computational compensation by the experimental study; this represents a novel technique to deal with the drawbacks of perturbations in the aerial manipulation systems.

*Email address: aaron.eleazar@inaoep.mx

The computational compensation strategy is based on the k-nearest neighbors (KNN) algorithm.

KNN method has attracted the attention of the research community due to its simplicity and effectiveness [11]. This technique has a wide range of applications such as density estimation, dimensional hashing, pattern recognition, data compression, and so on [12]. The nearest neighbor search is an optimization problem, whose goal is to find an instance that minimizes a certain distance or similarity function [11, 12]. In this work, we utilize the KNN algorithm to classify the noise induced by the movement of the arm of the manipulator in order to infer and send a compensation signal to the stability control of the aerial system. The main reason for the election of the KNN method is the requirement of a not a complicated training process to set up a classifier and only utilizes the labeled training set to classify the testing data.

This paper is organized as follow: In section 2 the novel proposed aerial manipulation system is described. The PID control technique and the compensation strategy developed for this work are presented in section 3. In section 4 the PID control with compensation is described. Three type of experiments were implemented to prove the effectiveness of the proposed strategy. Such experiments, and the obtained results are described in section 5. Finally, the main contribution, conclusions, and future direction are discussed in section 7.

2 DESCRIPTION OF PROPOSED SYSTEM

Among the different configurations of UAV systems, the VTOL vehicle has been taken into account specially for aerial manipulation due to their specific aspects in the flight mode. Particularly, the quadcopter is used in this work to achieve the primary goal of maintaining the robot manipulator in the desired point. In this work, an aerial manipulator is designed. The system consists of two primary subsystems: the aerial vehicle of four rotors and a robotic arm of two DOF. In the following subsections, each one of this system is presented.

2.1 FOUR ROTOR STRUCTURE

Due to the four rotors, the quadcopter has more lifting power than a helicopter of the same size, allowing to carry on a heavier payload. In Figure 1 a four-rotor structure model is shown, which corresponds to the physical structure of a parrot bebop-2. Also in this figure illustrated our own design of the two DOF arm attached to the bebop-2. The robotic arm was designed specifically for this aerial vehicle taking care of the dimensions and weights of each piece in order to assure most possibly the stability of the vehicle during the flight. Four extension legs were also designed to provide free taking off and landing of the aerial vehicle when carrying the arm.

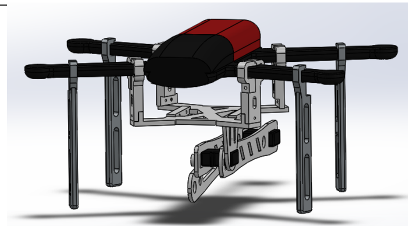


Figure 1: CAD of 2 DOF robotic arm.

2.2 ROBOTIC ARM

In the task of manipulation and interaction, robotic arms can provide the necessary degrees of freedom to achieve the objective [13, 2, 8]. In contact with the environment, for example, an n-DOF arm could supply the stiffness and versatility to the vehicle to accomplish the goal involving contact with a rigid structure. The N-DOF arm could also designed to provide a safe distance between the aerial system and the structure.

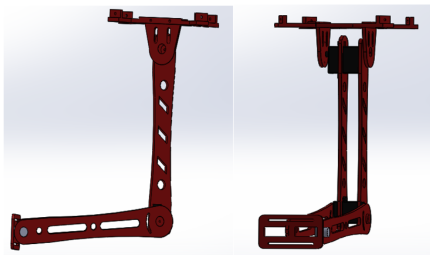


Figure 2: CAD of 2 DOF robotic arm.

Figure 2 shows our design of the two DOF-arm for the bebop-2 structure. The design was developed thinking in two main aspects. First, the physical task. The task consists in exert a force on a rigid surface. For this objective, the robotic arm must provide the necessaries movements to successfully contact the surface and also must provide adequate distance from the vehicle to the surface to reduce the disturbances induced by the proximity of the rigid structure with the rotors of the aerial vehicle. Secondly, the dimensions of the proposed design must maintain a relationship with the physical capabilities of the system guarantying the typical performance in the flight mode. Even with this consideration, the behavior of the robotic arm can alter the efficiency of the vehicle, for this reason, a control technique which considers the perturbations of the robotic arm must be implemented to achieve the proposed task. In the following sections, the problem of perturbations and the proposed solution is handled.

3 BACKGROUND

This section presents the background theory of the PID control and the KNN strategy, as a computational-based compensation strategy incorporated into the control loop.

3.1 PID CONTROL DESIGN

Considering that the dynamical model of the quadcopter is an under actuated, highly coupled and nonlinear system, a considerable number of control strategies have been developed for such class of similar systems [14, 15]. Among them, sliding mode control, which has drawn researchers' much attention, has been an useful and efficient control algorithm for handling systems with significant uncertainties, time-varying properties, nonlinearities, and bounded external disturbances. Most of the control strategies mentioned above have been proposed to help in the stability of the quadcopter on finite-time. A PID controller continuously calculates an error value $e(t)$ as the difference between the desired set point and a measured process variable and applies a correction based on proportional, integral, and derivative terms, (sometimes denoted P, I, and D respectively). The PID algorithm is described by:

$$u(t) = k_p e(t) + k_I \int e(\tau)d\tau + k_D \frac{d}{dt}e(t) \qquad (1)$$

In equation 1 k_p, k_I, k_D are the PID control gains, $u(t)$ is the control signal and $e(t)$ is the control error. The integral, proportional and derivative part can be interpreted as control actions based on the past, the present and the future. The PID gains can be designed based upon the system parameters if they can be achieved or estimated precisely. We mentioned next some related works. In 2010 Pual E. I. Pounds et al. [16] developed a dynamic model of an aerial manipulation system during object capture by combining a simple planar model of a helicopter UAV in hover under PID control with a suspended bogie linkage representation of a compliant gripper. Matko Orsarg et al. [17] proposed a proportional controller for the speed loop and a proportional integral controller for the position loop taking into account the dynamics of the system composed by a VTOL vehicle with a 2 DOF manipulator. A PID controller is designed for the x and y position and yaw orientation of the aerial vehicle.

3.2 K NEAREST NEIGHBOR COMPUTATIONAL COMPENSATION STRATEGY

KNN is a non-parametric method used for classification problems. The method requires the construction of a data base consisting of training examples, where each example $f(x)$ is represented by a set of n attributes $f = (a_1, a_2, ..., a_n$ and a corresponding class x. Once the data base is ready, the classification model consists of an input vector for which the class is unknown. The output is a class membership inferred by a majority vote of the k nearest examples found in the data base.

In sum, the KNN algorithm consists of two main phases: training and classification. The training phase comprises only of storing the feature vector examples and class labels of the training samples $< x, f(x) >$, where $x \in X$, where X is the set of all classes.

In the classification phase, an unlabeled vector $(\hat{f})$ is classified by assigning the label which is most frequent among the k training samples nearest to that query point; the following equation describes this phase of the KNN algorithm:

Therefore, in this work, our goal is to use the KNN algorithm as the means to generate a model that has knowledge on what control signal should be used to compensate for a disturbance. The next section will describe this proposed approach.

4 PID CONTROL ALGORITHM WITH COMPENSATION STRATEGY

We performed several runs where we associated a control signal u_c with the vehicle's motion disturbance f derived from the arm's motion. These disturbances were observed via the motion capture system in the manner of shifts of the vehicle's position P concerning to the set point in the air. In this manner, we created a database (see Table 1) containing the arm's disturbance coupled with the vehicle's shifts. We consider this a learning stage in our compensation approach.

f		P	U_c
$f(p_1)$	1	p_1	U_{c1}
	.	.	
	.	.	
$f(p_i)$	i	p_i	U_{ci}
	.	.	
	.	.	
$f(p_N)$	N	p_N	U_{cN}

Table 1: Database structure.

Once the database was created, a compensation scheme is implemented by inferring the nearest disturbance effect f in the vehicle's motion given an arm's disturbance. According to variation of real trajectory of the system a correction U_c necessary to get back the system to the desired trajectory is determined. Equation 2 represents the new control signal.

$$u(t) = k_p e(t) + k_I \int e(\tau)d\tau + k_D + \frac{d}{dt}e(t) + u_c \qquad (2)$$

This is, in a new flight test, we generate a disturbance with the arm's movement, and at the same time, we seek out the most similar disturbance recorded in our database in order to obtain a value representing the expected vehicle's shift position, this value will be used to compensate the signal in the PID controller of the vehicle's flight. For the implementation of the algorithm, we choose N= 10. In sum, we employ a 1-Nearest Neighbor approach, also known as lazy learning in computational terms, to infer a disturbance given a set of examples previously experienced. Our hypothesis is that, by inferring this disturbance, the PID controller will struggle less to maintain the vehicle's motion around the set point. In the following section, the development of the control technique is detailed.

5 EXPERIMENTAL SETUP

The proposed control strategy was proved via experimental tests in a controlled environment. We considered physical dimensions of bebop-2 to determine the appropriate design of the manipulator. The two degree of freedom (DOF) robotic arm is composed of 2 links with a dimension of 10 cm (L), 0.3 cm (W) and 4 cm (H), each one with 0.12 kg (m). An Arduino Nano board was used to control the manipulator.

Figure 3: Aerial Manipulation system implemented for this work.

The proposed system is shown in Figure (3), the CAD model shown in section 2.1 (a) was developed first to estimate the proper dimension of pieces, then the manipulator was built using 3D-print technology. Finally, all components were attached to the bebop-2 vehicle (b). The Figure 4 shows the block diagram for this research. In this representation $E(t)$ represents the error between the desired position and the real position, $Y(t)$ represents the control commands sent to the aerial vehicle to control the attitude. A PID control was implemented to each signal of the position of the system (x, y, z, yaw). The VICON cameras were used to get the position of the system continually.

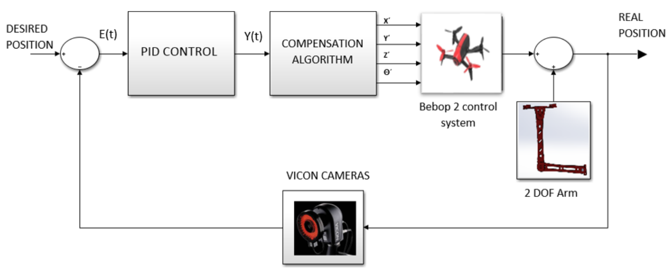


Figure 4: . Block diagram of the proposed system with compensation strategy.

The instability due to the arm is represented as noise in the block diagram that affects the position of the vehicle directly. The objective is to compensate this displacement in order to maintain the aerial vehicle in the desired position. The disturbance of the robotic arm is compensated by the PID control with a compensation signal.

To prove the necessity of a compensation control for the stability of the bebop-2, a case of study of the behavior of the system in hovering mode was done. Figure 5 show the shifted position of bebop-2 from desired point (0,0,1) due to the arm. The result illustrates the necessity to implement a control algorithm to help the inner control of the bebop-2 to maintain stability. Three experiments where implemented to prove the proposed control technique. Figure 6 shows the proposed movements of the robotic arm to implement the compensation strategy. The manipulator arm start in the initial position (1 and moves until the final position 4). In the following section, the results of these experiments are detailed.

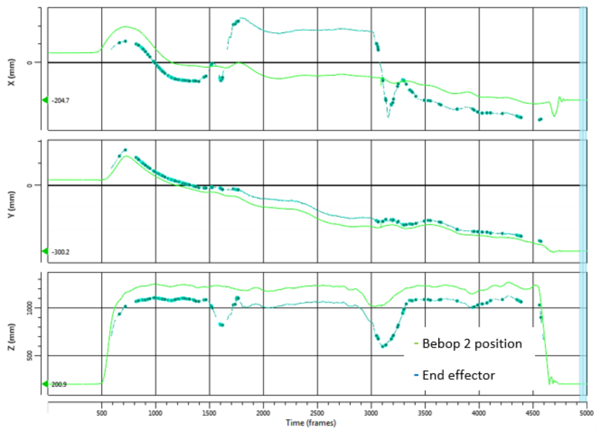


Figure 5: Hovering mode results. Behavior of the aerial vehicle carrying the robotic with movement.

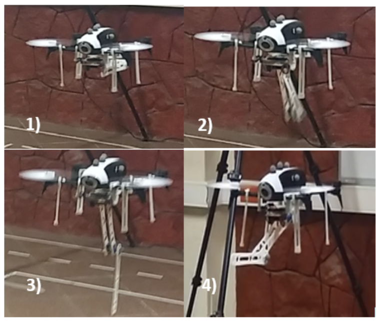


Figure 6: First sequences of Movements of the robotic arm during flight mode.

6 RESULTS

In the first experiment, the PID control 3.1 is implemented to maintain the aerial manipulator at the desired point; the arm does not move during this experiment. The graphics of

Figure 7 shows the position of the system in x-axis and y-axis, yaw orientation is also included. The PID control is trying to maintain the vehicle in position $(4, 3, 1)$. The x-axis of the three graphics represents the position in millimeters (mm) and the y-axis of the graph represents the time in milliseconds (ms). In this example, the PID control maintains the system among (0.2, -0.2) in all axes.

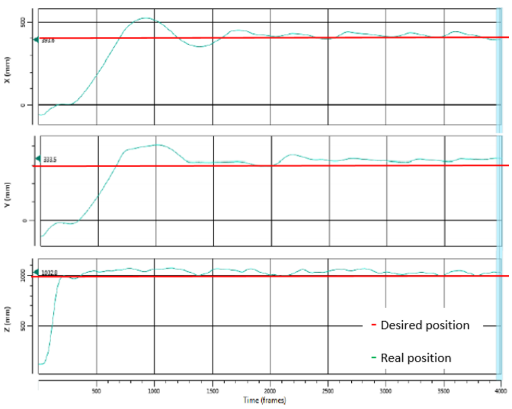


Figure 7: PID control results. Behavior of the aerial vehicle carrying the robotic arm but the arm is not moving.

In the second experiment, the arm executes predefined movements as described in Figure 6, in this experiment the PID control is incapable of ensuring the stability of the system at the desired point. The error $e(t)$ is above 8 cm. Bebop 2 is located in position (0, 0, 1) when the robotic arm starts to move. Two blue fringes highlight the lapse time where the arm is moving, left fringe indicates the movement of the arm from initial position $(1$ to final position $(4$ and right fringe indicate the movement of the arm to return to initial position. Figure 8 shows that the displacement of the system continues even when the arm stops moving.

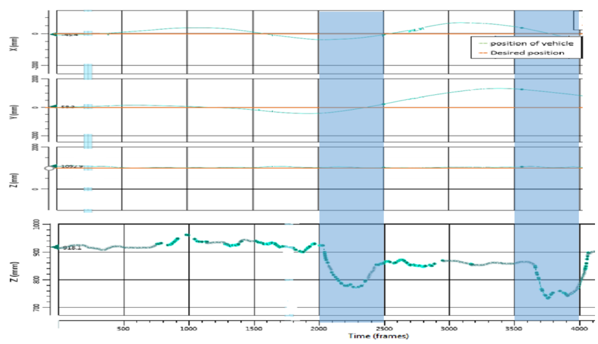


Figure 8: PID control results. Behavior of the aerial vehicle carrying the robotic arm and the arm is moving.

To improve the effectiveness of the PID control, the compensation strategy described in section 3.2 is implemented. Now the PID control is able to compensate for the disturbance produced by the arm. The graphics of Figure 9 shows the position of the system in the x-axis, y-axis, and yaw orientation. When comparing graphics of Figure 8 and Figure 9, the effectiveness of the PID control with compensation strategy is demonstrated.

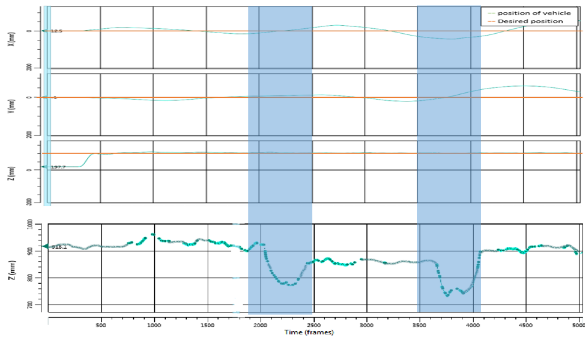


Figure 9: PID control with compensation results. Behavior of the aerial vehicle carrying the robotic arm and the arm is moving

For the purpose of quantitatively compare the control performance we defined the following function:

$$mse = \frac{1}{n}\sum_{i=1}^{n}(e_i)^2 \tag{3}$$

Equation 3 describe the mean squared error (mse) where $e_i = p_d - p_r$ represents the error value between desired position p_d and real position p_r of the system in the time i. The quantitative results for each controller are given as in Table 2. According to the experimental results, the disturbance induced by the arm seems to affect position x more than in the others positions, this is owing to the arm moves in the x plane and the displacement in the other positions is the result of the adjustment of the control in the rotors of the vehicle. Comparing results we observe that an upgrade in the performance and response has achieved with PID control with compensation.

mse	Hovering	PID control	PID + comp.
X	42.56	14.12	5.25
Y	26.44	11.45	2.98
Z	7.65	4.95	1.62
θ	4.2	2.6	1.02

Table 2: Mean squared error.

7 CONCLUSIONS

We have presented a novel aerial manipulation system. Our novel design is affordable and can be produced via 3D printing technology. Experimental studies were made with the aerial manipulator developed in this work to determine physically the variation of dynamics and deterioration of the

performance of the air vehicle due to the incorporated arm. We have also presented preliminary results regarding a control strategy to maintain stable flight of the vehicle while the arm performs a task. The strategy involves the use of a computational approach based on soft learning, where a training stage is carried out in order to create a database confirmed by examples of disturbances induced on the vehicle and derived from the arm's motion, and we related them to the control signals sent to the arm. Thus, in the test stage, when we sent a control signal to the arm, we infer from the learned database the vector disturbance that may arise. After that, we use such information to compensate for the disturbance by adding a correction signal to the PID controller calculated in the experimental study phase. This represents a novel technique to deal with the drawbacks of perturbations in the aerial manipulation systems. The main contribution of this work is the novel control technique based on a traditional PID algorithm with a compensation strategy and the two- DOF arm to the aerial vehicle.
Link of video: https://youtu.be/2BB0aDr6-lQ

References

[1] C. D. Bellicoso, L. R. Buonocore, V. Lippiello, and B. Siciliano. Design, modeling and control of a 5-dof light-weight robot arm for aerial manipulation. In *Control and Automation (MED), 2015 23th Mediterranean Conference on*, pages 853–858, June 2015.

[2] S. Bouabdallah and R. Siegwart. Full control of a quadrotor. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 153–158, Oct 2007.

[3] D. Bazylev, A. Kremlev, A. Margun, and K. Zimenko. Design of control system for a four-rotor uav equipped with robotic arm. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2015 7th International Congress on*, pages 144–149, Oct 2015.

[4] T. Bartelds, A. Capra, S. Hamaza, S. Stramigioli, and M. Fumagalli. Compliant aerial manipulators: Toward a new generation of aerial robotic workers. *IEEE Robotics and Automation Letters*, 1(1):477–483, Jan 2016.

[5] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar. Grasping from the air: Hovering capture and load stability. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2491–2498, May 2011.

[6] M. Bernard and K. Kondak. Generic slung load transportation system using small size helicopters. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3258–3264, May 2009.

[7] K. Baizid, G. Giglio, F. Pierri, M. A. Trujillo, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero. Behavioral control of unmanned aerial vehicle manipulator systems. *Autonomous Robots*, 41(5):1203–1220, Jun 2017.

[8] L. R. Buonocore, J. Cacace, and V. Lippiello. Hybrid visual servoing for aerial grasping with hierarchical task-priority control. In *Control and Automation (MED), 2015 23th Mediterranean Conference on*, pages 617–623, June 2015.

[9] Fabio Ruggiero, Jonathan Cacace, Hamid Sadeghian, and Vincenzo Lippiello. Passivity-based control of {VToL} {UAVs} with a momentum-based estimator of external wrench and unmodeled dynamics. *Robotics and Autonomous Systems*, 72:139 – 151, 2015.

[10] F. Santoso, M. A. Garratt, and S. G. Anavatti. State-of-the-art intelligent flight control systems in unmanned aerial vehicles. *IEEE Transactions on Automation Science and Engineering*, 15(2):613–627, April 2018.

[11] J. Zhang, M. Gao, W. Chen, and G. Shen. Non-data-aidedk-nearest neighbors technique for optical fiber nonlinearity mitigation. *Journal of Lightwave Technology*, 36(17):3564–3572, Sept 2018.

[12] Jorge Gonzalez-Lopez, Sebastin Ventura, and Alberto Cano. Distributed nearest neighbor classification for large-scale multi-label data on spark. *Future Generation Computer Systems*, 87:66 – 82, 2018.

[13] S. Kannan, M. Alma, M. A. Olivares-Mendez, and H. Voos. Adaptive control of aerial manipulation vehicle. In *Control System, Computing and Engineering (ICCSCE), 2014 IEEE International Conference on*, pages 273–278, Nov 2014.

[14] C. E. Doyle, J. J. Bird, T. A. Isom, C. J. Johnson, J. C. Kallman, J. A. Simpson, R. J. King, J. J. Abbott, and M. A. Minor. Avian-inspired passive perching mechanism for robotic rotorcraft. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4975–4980, Sept 2011.

[15] S. Tang and V. Kumar. Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2216–2222, May 2015.

[16] Aaron Dollar Paul E. I. Pounds. Hovering sability of helicopters with elastics constraints. *ASME Dynamic Systems and Control Conference*, 2010.

[17] Matko Orsag, Christopher Korpela, and Paul Oh. Modeling and control of mm-uav: Mobile manipulating unmanned aerial vehicle. *Journal of Intelligent & Robotic Systems*, 69(1):227–240, 2013.

Performance of Unmanned Aircrafts in the Event of Loss-of-control

Torbjørn Cunis* and Elgiz Baskaya†
French Civil Aviation School, 31055 Toulouse, France

Abstract

Loss of control is a severe and immediate consequence of faults in an unmanned aircraft system during flight. Without recovery, detection of admissible landing spots is necessary to avoid causalities. This case study considers exemplary the MAKO unmanned aircraft and discusses viable trim conditions for a drone system after faults of propulsion and elevator using continuation analysis. Furthermore, simple estimates of reachable zones for controlled flight into terrain are provided.

1 Introduction

As the variety of drone application widens and stakeholder push for further utilisation of the airspace, aviation authorities are busy today with new regulations for unmanned aircraft systems. in order to set a common ground for European aviation and integrate drones safely into the airspace. Due to both decreased cost and accuracy, drone systems are often susceptible to faults of different nature. To achieve a safe integration into the airspace, the problem of detecting anomalies during flight has to be addressed. Taking advantage of intelligent software which processes available information as efficient as possible could serve as an effective way to contribute to the solution of this challenging problem.

Aircraft *loss of control* (LOC) is considered as one of the biggest contributors to fatal accidents. The Joint Safety Analysis Team (JSAT) defines LOC as a "significant, unintended departure of the aircraft from controlled flight, the operational flight envelope, or usual flight attitudes, including ground events" [1], where *significant* refers to events resulting with an incidence or accident. Control failures, inappropriate pilot action (or simply inaction) in a healthy aircraft, and vehicle impairment are examples of LOC events [2]. In-flight loss of control (LOC-I) in particular is the most deadly accident type with 37 fatal accidents per year[1] [3]. Although LOC-I is the cause behind many fatal accidents, manned aviation has a very limited use of LOC prevention and recovery [4]. Having no single action to prevent LOC events, technical limits in LOC simulations such as full stall or failure simulations, constitute some of the challenges on the way to LOC event prevention and recovery solutions. LOC events have been categorized by [5] under five main topics, the most common being aerodynamic stall and flight control system failures. The category of flight control failures, the second most fatal cause of LOC in 15 years log, includes autopilot commands and control surface failures [5]. The study further shows that the percentage of flight control system upset incidents among the other LOC events has risen from 9 % to 22 % after 1993 (pre-1993 compared to 1993–2007). With increased complexity of onboard systems, addressing onboard fault detection and recovery could contribute to reduce the likelihood of LOC accidents and their fatalities.

Designing recovery measures for unmanned systems has further challenges due to the lack of redundancies and use of cheaper and less accurate components compared to manned aviation. Here, fault tolerant control systems (FTCS) are designed to issue solutions to systems which are under fault/failure. There are a wide range of different strategies offered for this solution such as passive or active FTCS, where the latter requires a fault detection and diagnosis (FDD) phase [6]. After the fault is known, the severity of the situation and current abilities of the drone need to be evaluated to decide if a recovery is possible. Recent studies considered viable trim conditions [7, 8], recoverability [9], and control invariant sets [10, 11] of transport aircrafts in case of impairment, reduced efficiency due to icing, or reduced control authority. In case a recovery is likely to fail, a safe ditch manoeuvre can abruptly decrease the number of fatalities. Maps pointing zones with no or minimum population could be uploaded onboard and the safest region to ditch can be selected. Since those situations are usually handled by aircraft pilots assessment and planning of ditch manoeuvres is imperative for the development of unmanned systems. NASA offers *Safe2Ditch* [12] to offer autonomous crash management yet is at design stage now.

In this paper, we provide a case study for discussion of abilities of a drone under failure. We consider an already detected fault of a single control input of the longitudinal aerodynamics – rather than a gradually degradation or impairment of a surface – namely loss of propulsion, loss of elevator, and loss of single aileron, and assess the remaining abilities of the drone including the calculation of possible landing zones in the first two cases. Throughout the paper we assume that vertical control surfaces exist and remain unimpaired.

*Doctoral researcher, ONERA – The French Aerospace Lab, Department of Information Processing and Systems; associated researcher with the French Civil Aviation School, Drones Research Group; e-mail: torbjoern.cunis@onera.fr

†Doctoral researcher, French Civil Aviation School, Drones Research Group; e-mail: elgiz.baskaya@enac.fr

[1] for fixed-wings.

2 MAKO Unmanned Aircraft

The MAKO unmanned aircraft shown in Fig. 1 has been selected as the drone to be modeled in the course of this study. MAKO has an off-the-shelf frame and is equipped with a single, back-facing propeller and two vertical control surfaces. Elevator and aileron of MAKO are combined in a single horizontal surface on each wing, called *elevon*. The frame is designed to be lightweight and easy to repair, and for that reason made from Elapor foam which however is limiting for the structural strength. The specifications of the airframe are given in Tab. 1. Stability derivatives for the aerodynamic forces and moments has been calculated via AVL and given in the appendix (Table 2). AVL is an open source program developed at MIT and uses vortex-lattice method for the aerodynamic and stability calculations. The output of the program is linearised at a selected condition, therefore all the coefficients are calculated around the equilibrium point at 14 m/s cruise flight condition. The centre of gravity is located at $x_{CG} = 0.295\,\mathrm{m}$, which corresponds to a 8 % positive static margin evaluatated in flight.

Table 1: Parameters of the MAKO aircraft [13].

flight mass	m	0.7–2.0	kg
wing span	b	1.29	m
mean chord	c_A	0.21	m
wing area	S	0.27	m^2
propeller diameter	D	22.8	cm
air density	ϱ	1.27	kg/m
gravitational constant	g	9.81	m/s^2

3 Nonlinear Model

We refer to the axis systems of the international standard: the *body axis system* aligned with fuselage; the *air-path axis system* aligned with the air-path velocity vector $\mathbf{V}_\mathrm{A}$; and the *normal earth-fixed axis system*. [14]

3.1 Equations of motion

Consider the ordinary differential equations

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{1}$$

with the state vector $\mathbf{x} = \begin{bmatrix} V_\mathrm{A} & \gamma_\mathrm{A} & q & \Theta \end{bmatrix}^T$ with airspeed V_A, inclination γ_A, pitch rate q, and pitch angle Θ; and the inputs $\mathbf{u} = \begin{bmatrix} \eta & n \end{bmatrix}^T$ comprised of elevator deflection η and engine speed n. The nonlinear equations of motion are given with respect to the states as

$$m\dot{V}_\mathrm{A} = F\cos\alpha - \frac{1}{2}\varrho V_\mathrm{A}^2 S C_\mathrm{D}(\alpha, q, \eta) + mg\sin\gamma_\mathrm{A}; \tag{2}$$

$$mV_\mathrm{A}\dot{\gamma}_\mathrm{A} = F\sin\alpha + \frac{1}{2}\varrho V_\mathrm{A}^2 S C_\mathrm{L}(\alpha, q, \eta) - mg\sin\gamma_\mathrm{A}; \tag{3}$$

$$I_y\dot{q} = \frac{1}{2}\varrho V_\mathrm{A}^2 S c_\mathrm{A} C_\mathrm{m}(\alpha, q, \eta); \tag{4}$$

$$\dot{\Theta} = q; \tag{5}$$

Figure 1: Photograph of the MAKO unmanned aircraft.

and the angle of attack $\alpha = \Theta - \gamma$, where C_L, C_D, and C_m denote the aerodynamic lift, drag, and pitch-moment coefficients, respectively. The thrust force is modeled as function of engine speed and airspeed,

$$F(n) = \varrho n^2 D^4 \left(C_{\mathrm{F}0} + C_{\mathrm{F}J}\frac{V_\mathrm{A}}{nD} + C_{\mathrm{F}n} n \right). \tag{6}$$

3.2 Aerodynamic coefficients

Using AVL, we obtain the aerodynamic coefficients for small angles of attack:

$$C_\mathrm{L} = C_{\mathrm{L}0} + C_{\mathrm{L}\alpha}\alpha + C_{\mathrm{L}q}\hat{q} + C_{\mathrm{L}\eta}\eta; \tag{7}$$

$$C_\mathrm{D} = C_{\mathrm{D}0} + C_{\mathrm{DL}}C_\mathrm{L}^2(\alpha, \hat{q}, \eta); \tag{8}$$

$$C_\mathrm{m} = C_{\mathrm{m}0} + C_{\mathrm{m}\alpha}\alpha + C_{\mathrm{m}q}\hat{q} + C_{\mathrm{m}\eta}\eta; \tag{9}$$

with the normalized pitch rate $\hat{q} = c_\mathrm{A} q / V_\mathrm{A}$. We further identified the stall angle of attack in an AVL subroutine [15] to

$$\alpha_\mathrm{stall} = 11.3°$$

and extend (7) to

$$C_\mathrm{L}' = C_\mathrm{L}(\alpha, \hat{q}, \eta) - \frac{1}{2\alpha_\mathrm{stall}} C_{\mathrm{L}\alpha}\alpha^2 \tag{10}$$

such that the lift force decreases beyond stall.

4 Analysis

The flight envelope is characterised by the stable trim conditions such that the aircraft satisfies given constraints on the states and limitations on the control inputs. Here, we require $\gamma \in [-30°; 30°]$ and $\alpha \in [-3°; 12°]$, denoted by $\mathbf{x} \in \mathcal{C}$, as well as $\mathcal{U} = [-10°; 10°] \times [0; 125\,\mathrm{rev/s}]$. If the control authority is compromised, some trim conditions are unachievable, thus limiting the abilities of the drone to fly stably.

The nonlinear system of (1) is in a viable trim condition if

$$\mathbf{f}(\mathbf{x}^*, \mathbf{u}^*) = 0 \tag{11}$$

for $\mathbf{x}^* \in \mathcal{C}$, $\mathbf{u}^* \in \mathcal{U}$. A trim condition is further stable if and only if the Jacobian matrix,

$$\mathbf{J}^* = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}^*, \mathbf{u}^*), \tag{12}$$

has only strictly negative eigenvalues. If, for some input $\mathbf{u}^*$, an eigenvalue of the Jacobian at the corresponding trim condition $(\mathbf{x}^*, \mathbf{u}^*)$ crosses zero and the trim condition changes stability, we have a critical point. For a single input parameter u, trim analysis is the result of a continuation method, where the equalities of $\mathbf{f}(\mathbf{x}, u^*) = 0$ are solved for small changes in u^* and stability of each trim condition is determined by $\mathbf{J}^*$. Continuation and bifurcation of a given system can be computed using mathematical toolboxes such as [16–18].

4.1 Loss of propulsion

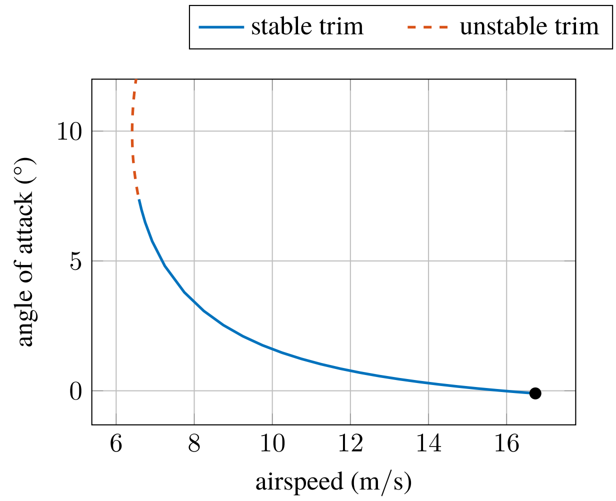
Without a propulsion system, the aircraft is not able to input energy. That is, if the speed is to be constant, the altitude must decrease, thus exchanging potential for kinetic energy. In consequence, we expect a viable trim condition at zero thrust to have a negative inclination angle.

Fig. 2 shows the viable trim conditions in the event of loss of propulsion. The largest inclination is achieved for an elevator deflection $\eta \approx 2.8°$ with airspeed $V_\mathrm{A} \approx 7.7\,\mathrm{m/s}$. For larger deflections, the aircraft enters into a steeper descent thus speeding up; for negative deflections, however, angle of attack is increased until the trim condition becomes unstable and finally, the wing stalls ($V_\mathrm{stall} \approx 6.4\,\mathrm{m/s}$). With airspeed and inclination at stable trim condition given by the elevator deflection, we can compute the components of the velocity vector, that is, the horizontal and vertical speed. The relation of horizontal and vertical motion is illustrated in Fig. 3. Unsurprisingly, the steep descent provides high speeds in both components; whereas close to stall, the vertical descent is amplified with respect to the horizontal speed.

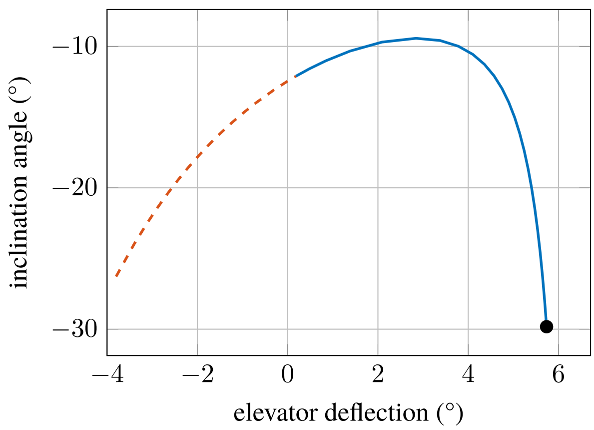
4.2 Loss of elevator

If the elevator surface is jammed, the aircraft looses authority over its main control surface of the longitudinal attitude. The position η_0 of the stuck surface now determines immediately the angle of attack at which the aircraft is trimmed. However, as the remaining throttle input enters the system in (2) and (3), there is still the possibility to control – and possibly stabilise – speed and inclination of the vehicle's path.

Trim conditions in the case of loss of elevator are shown by Fig. 4 for surface positions of $-1°$, $0°$, $1°$, and $1.5°$; the corresponding angles of attack are $9.0°$, $7.6°$, $6.3°$, and $5.6°$, respectively. Fig, 4 also shows the critical boundary, i.e., the boundary of stable trim conditions, which is a function of the position of the stuck elevator surface. The viable trim conditions are further limited by the necessity of a positive throttle command ($n \geq 0$), since there is no negative thrust. Inputs that prompt a stable trim condition are shown by Fig. 5. As we can obtain from this trim analysis, the MAKO aircraft is barely stable in case of an elevator jam in neutral position and neither for negative (pitching upwards) deflections. For increasingly positive deflections however, larger ranges of engine speed yield a stable trim.



(a) Angle of attack α over airspeed V_A.



(b) Inclination γ over elevator deflection η.

Figure 2: Viable trim conditions in the event of loss of propulsion.

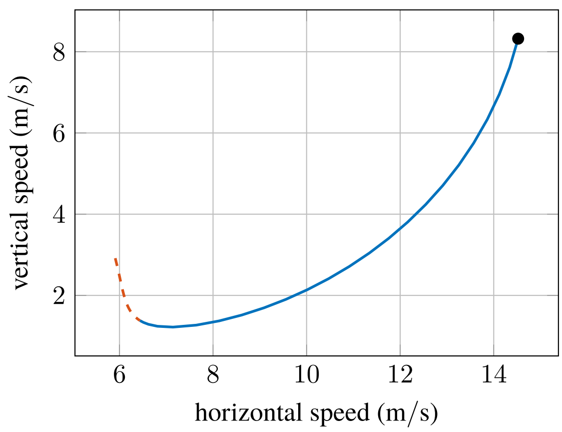
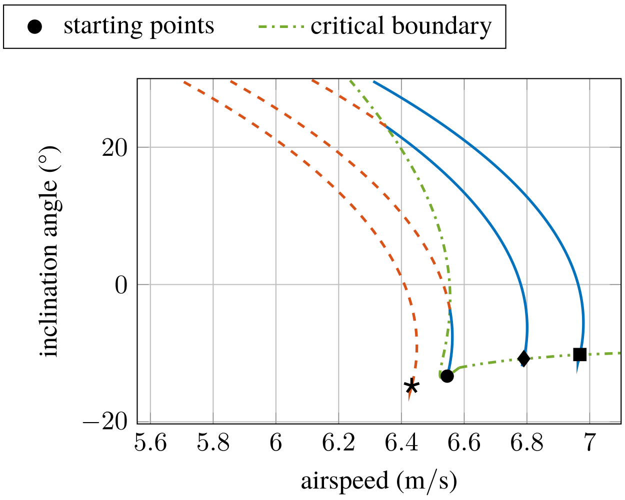
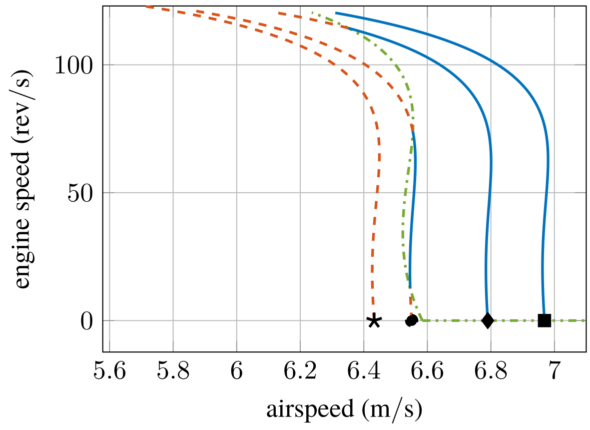


Figure 3: Composition of the velocity vector at trim condition in the event of loss of propulsion.



(a) Inclination γ over airspeed V_A.



(b) Engine speed n over airspeed V_A.

Figure 4: Viable trim conditions in the event of loss of elevator. (Surface position: $\star$ $\eta_0 = -1°$; $\bullet$ $\eta_0 = 0°$; $\blacklozenge$ $\eta_0 = 1°$; $\blacksquare$ $\eta_0 = 1.5°$.)

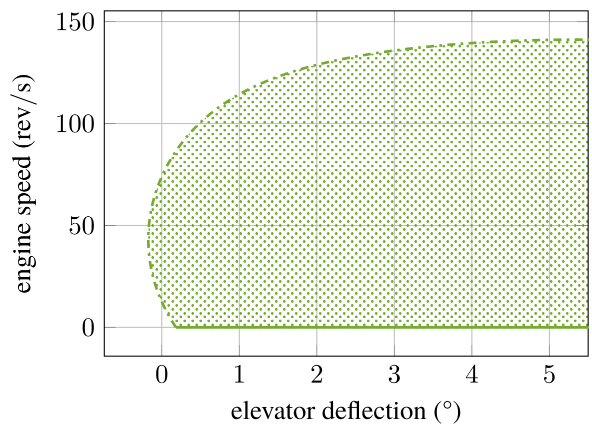


Figure 5: Stable inputs in the event of loss of elevator: engine speed n over surface position η_0.

4.3 *Loss of single aileron*

A single aileron jammed in no neutral position induces a roll moment onto the aircraft, making trimming without further relaxations difficult. If the second aileron can be deflected independently and to a symmetric – rather than anti-symmetric as usual – position equating the stuck surface, this cancels the roll moment and the discussion of trim conditions is similar to the previous case (for MAKO, where elevator and ailerons are combined into the elevon surface, we actually have the identical case). On the other hand, a relaxation on the trim constraints such as a nonzero roll rate might also lead to a deteriorated yet stable flight condition enabling a controlled descend.

A discussion of relaxed trim conditions requires an augmentation of the dynamic system with the lateral, six-degrees-of-freedom equations of motion as well as the continuation of limit cycles and is beyond the scope of this paper.

5 Landing Zones

If the aircraft cannot continue its mission due to an impairment, a suitable spot for landing or controlled flight into terrain is crucial in order to avoid further accidents. When discussing landing spots, we obvious have to take into account the vehicle's position $\mathbf{x}_g$ in the earth-fixed reference frame; we thus extend the state vector to $\mathbf{y} = \begin{bmatrix}\mathbf{x} & \mathbf{x}_g\end{bmatrix}^T$ with

$$\dot{x}_g = V_A \cos\gamma_A \tag{13}$$

$$\dot{z}_g = -V_A \sin\gamma_A \tag{14}$$

representing the horizontal and vertical position. We denote the extended system dynamics by $\bar{\mathbf{f}}$ and have the constraints $\bar{\mathcal{C}} = \mathcal{C} \times \{\mathbf{x}_g \,|\, z_g \le 0\}$. Given a certain aerodynamic state and position $(\mathbf{y}_0)$, we then have the set of *admissible landing spots* as

$$\begin{aligned}\mathcal{LS}[\mathbf{y}_0] = \{\mathbf{x}'_g \,|\, \exists \mathbf{x}' \in \mathcal{C}_L,\ \mathbf{y}(\cdot) \subset \bar{\mathcal{C}},\ \mathbf{u}(\cdot) \subset \mathcal{U},\ T > 0. \\ \dot{\mathbf{y}} = \bar{\mathbf{f}}(\mathbf{y}, \mathbf{u}), \mathbf{y}(0) = \mathbf{y}_0, \mathbf{y}(T) = \begin{bmatrix}\mathbf{x}' & \mathbf{x}'_g\end{bmatrix}^T\} \\ \cap\, \mathfrak{Surf},\end{aligned} \tag{15}$$

where $\mathcal{C}_L \subset \mathcal{C}$ are the landing constraints and $\mathfrak{Surf}$ is a model of the surface. For simplicity, we take $\mathfrak{Surf} = \{\mathbf{x}_g \,|\, z_g = 0\}$.

Solving (15) imposes an INF MIN optimal control problem. However, with some simplifications we can easily estimate bounds for a zone of controlled flight into terrain. Here, we relax the landing constraints to $\mathcal{C}_L = \mathcal{C}$ and further consider only trajectories where the aircraft is in trim condition almost all time. We further require that the set of stable, viable trim conditions is compact and connected.[2] Starting from the position difference

$$\Delta\mathbf{x}_g(\mathbf{y}_0, \mathbf{u}(\cdot)) = \int_0^T \dot{\mathbf{x}}_g(t)\,dt \tag{16}$$

[2]We will argue that this requirement is given for the discussed events of loss-of-control.

with $\dot{\mathbf{y}} = \bar{\mathbf{f}}(\mathbf{y}, \mathbf{u})$ and $\mathbf{y}(0) = \mathbf{y}_0$, we approximate (16) as

$$\Delta \mathbf{x}_{\mathrm{g}}(\mathbf{y}_0, \mathbf{u}(\cdot)) \approx \int_0^T \dot{\mathbf{x}}_{\mathrm{g}}^*(t)\,\mathrm{d}t \tag{17}$$

with $\bar{\mathbf{f}}(\mathbf{y}^*, \mathbf{u}) = \begin{bmatrix} 0 & \dot{\mathbf{x}}_{\mathrm{g}}^* \end{bmatrix}^T$ for almost all $t \in [0, T]$ and define the bounds

$$\overline{\Delta \mathbf{x}_{\mathrm{g}}} = T\dot{\mathbf{x}}_{\mathrm{g}}^*, \qquad \mathbf{f}(\mathbf{x}^*, \mathbf{u}_{\min}) = 0; \tag{18}$$

$$\underline{\Delta \mathbf{x}_{\mathrm{g}}} = T\dot{\mathbf{x}}_{\mathrm{g}}^*, \qquad \mathbf{f}(\mathbf{x}^*, \mathbf{u}_{\max}) = 0; \tag{19}$$

where $\mathbf{u}_{\min}$ ($\mathbf{u}_{\max}$) denote the respective control input such that there is a viable, stable trim condition with $\gamma_{\mathrm{A}} < 0$ and $|\gamma_{\mathrm{A}}|$ is minimal (maximal). We then have by boundedness of the interval (17)

$$\begin{aligned} \mathcal{CFT}[\mathbf{x}_{\mathrm{g}0}] = \big\{ \mathbf{x}_{\mathrm{g}}' \big| \exists \mathbf{u} \in [\mathbf{u}_{\min}; \mathbf{u}_{\max}], T > 0. \\ \mathbf{x}_{\mathrm{g}}' - \mathbf{x}_{\mathrm{g}0} = T\dot{\mathbf{x}}_{\mathrm{g}}^*, \mathbf{f}(\mathbf{x}^*, \mathbf{u}) = 0 \big\} \\ \cap\, \mathfrak{Surf} \end{aligned} \tag{20}$$

and obtain

$$\mathbf{x}_{\mathrm{g}} \in \mathcal{CFT}[\mathbf{x}_{\mathrm{g}0}] \Longleftrightarrow \frac{|x_{\mathrm{g}} - x_{\mathrm{g}0}|}{z_{\mathrm{g}0}} \in \left[\frac{\underline{\Delta x_{\mathrm{g}}}}{\underline{\Delta z_{\mathrm{g}}}}; \frac{\overline{\Delta x_{\mathrm{g}}}}{\overline{\Delta z_{\mathrm{g}}}} \right] \tag{21}$$

for $z_{\mathrm{g}} = 0$ if $\mathbf{x}_{\mathrm{g}} \in \mathfrak{Surf}$.

Landing without propulsion In the event of loss-of-propulsion, as we have seen in the section before, the aircraft has still a large range of stable yet descending trim conditions and the set of viable trim conditions is connected (Figs. 2 and 3). Therefore, the admissible zone of controlled flight into terrain depends mainly onto the initial height of the vehicle. Fig. 6 shows estimates for a linear approach and initial heights of 50 m, 150 m, and 250 m. Due to the fact that the aircraft is not able to descend vertically, there is not only a maximum reachability but also a minimal; clearly, the aircraft might also be able to reduce height in a descending turn, thus reducing the minimum distance for the (final) approach.

Landing with elevator jam Deprived of its mean of attitude control, the aircraft is in a severe state of fault; on the other, contrary to the case before, it still has the possibility to propel its flight – given sufficient battery or fuel. Whereas for the elevator jammed in neutral position there are only stable trim conditions with a descending path, for positive deflections the aircraft is too able to ascend stably (Fig. 4). In the latter case, the aircraft thus could, theoretically, stay aloft indefinitely: for instance, Fig. 7 illustrates the estimated admissible zones for controlled flight into terrain with elevator jams at deflections of 0°, 1°, 2°, and 5° (from quadrant I to IV). Note that a deflection of 5° without any throttle corresponds to fast and steep descent (see also Fig. 2) thus reducing the minimum approach distance with respect to the moderate deflections of 0° to 2°.

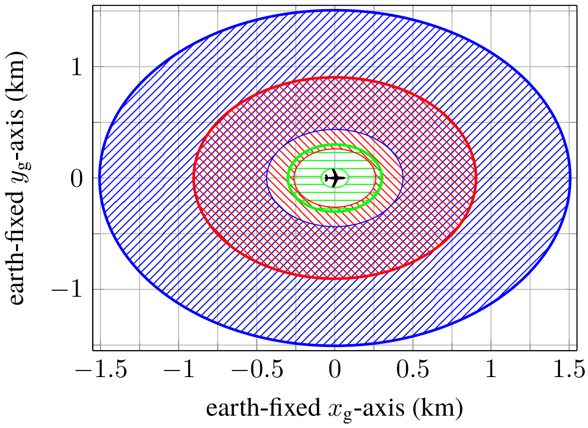


Figure 6: Estimated CFT zones for loss-of-propulsion. (Initial height: 50 m; 150 m; 250 m.)

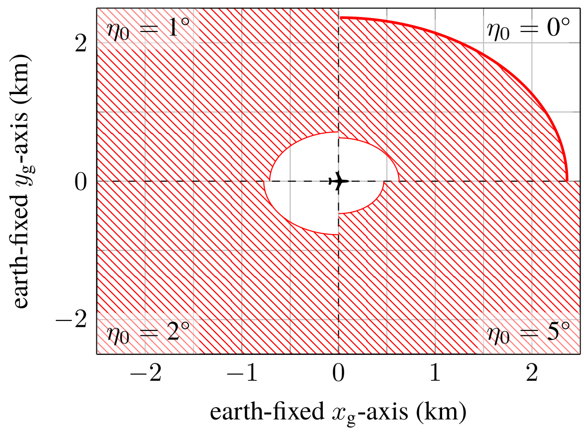


Figure 7: Estimated CFT zones for loss-of-elevator with initial height of 150 m () and surface positions as indicated.

6 Conclusion

If a drone loses control authority over its inputs, stable flight conditions are often significantly restricted and continuation of flight might not be possible at all. In order to avoid LOC, information about flight abilities is crucial, as well as reachable zones for controlled flight into terrain. In this study, we discussed viable trim conditions of the longitudinal flight for the MAKO unmanned aircraft in the events of loss of propulsion or elevator. While, without thrust, the aircraft remains stable yet descends, for the elevator being jammed in pitch-up or neutral positions stable flight is (almost) impossible. Consequently, reachable landing spots are restricted both by the possible, minimal and maximal negative inclination angles. The proposed approach can be evaluated during design of the aircraft system.

References

[1] P. Russell and J. Pardee. Joint safety analysis team-CAST approved final report loss of control JSAT results

and analysis. Technical report, Commercial Aviation Safety Team, Washington, US-DC, 2000.

[2] Nathan D. Richards, Neha Gandhi, Alec J. Bateman, David H. Klyde, and Amanda K. Lampton. Vehicle upset detection and recovery for onboard guidance and control. *Journal of Guidance, Control, and Dynamics*, 40(4):920–933, 2016.

[3] *Loss of Control (LOC-I)*. European Aviation Safety Agency, Cologne, DE, 2018. URL https://www.easa.europa.eu/easa-and-you/general-aviation/flying-safely/loss-of-control#group-easa-downloads. (accessed August 20, 2018).

[4] Christine M. Belcastro, John V. Foster, Gautam H. Shah, Irene M. Gregory, David E. Cox, Dennis A. Crider, Loren Groff, Richard L. Newman, and David H. Klyde. Aircraft loss of control problem analysis and research toward a holistic solution. *Journal of Guidance, Control, and Dynamics*, 40(4):733–775, 2017.

[5] A. Lambregts, Gregg Nesemeier, Richard Newman, and James Wilborn. Airplane upsets: Old problem, new issues. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, page 6867, 2008.

[6] Guillaume J.J. Ducard. *Fault-tolerant flight control and guidance systems: Practical methods for small unmanned aerial vehicles*. Springer, Berlin, DE, 2009.

[7] Harry G Kwatny, Jean-Etienne T Dongmo, Bor-Chin Chang, Gaurav Bajpai, Murat Yasar, and Christine Belcastro. Nonlinear Analysis of Aircraft Loss of Control. *Journal of Guidance, Control, and Dynamics*, 36(1): 149–162, 2013.

[8] Stefan Schuet, Thomas Lombaerts, Diana Acosta, John Kaneshige, Kevin Wheeler, and Kimberlee Shish. Autonomous Flight Envelope Estimation for Loss-of-Control Prevention. *Journal of Guidance, Control, and Dynamics*, 40(4), 2017.

[9] Kevin McDonough and Ilya Kolmanovsky. Fast Computable Recoverable Sets and Their Use for Aircraft Loss-of-Control Handling. *Journal of Guidance, Control, and Dynamics*, 40(4):934–947, 2017.

[10] Robert C. Allen, Harry G. Kwatny, and Gaurav Bajpai. Safe Set Protection and Restoration for Unimpaired and Impaired Aircraft. In *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, US-MN, 2012.

[11] H. N. Nabi, T. Lombaerts, Y. Zhang, E. van Kampen, Q. P. Chu, and Coen C. de Visser. Effects of Structural Failure on the Safe Flight Envelope of Aircraft. *Journal of Guidance, Control, and Dynamics*, 41(6):1257–1275, 2018.

[12] *Safe2Ditch Technology*. NASA Langley Research Center, Hampton, US-VA, 2018. URL https://technology.nasa.gov/patent/LAR-TOPS-243. (accessed August 22, 2018).

[13] Murat Bronz, Hector Garcia de Marina, and Gautier Hattenberger. In-flight thrust measurement using on-board force sensor. In *AIAA Atmospheric Flight Mechanics Conference*, page 0698, 2017.

[14] *Flight dynamics – Concepts, quantities and symbols – Part 1: Aircraft motion relative to the air*. International Organization for Standarization, Genève, CH, 4th edition, 1988.

[15] Murat Bronz. *A Contribution to the Design of Long Endurance Mini Unmanned Aerial Vehicles*. PhD thesis, Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse, FR, 2012.

[16] Harry Dankowicz and Frank Schilder. *Recipes for Continuation*. Society for Industrial and Applied Mathematics, Philadelphia, US-PA, 2013.

[17] A. Dhooge, W. Govaerts, Yu. A. Kuznetsov, H. G.E. Meijer, and B. Sautois. New features of the software MatCont for bifurcation analysis of dynamical systems. *Mathematical and Computer Modelling of Dynamical Systems*, 14(2):147–175, 2008.

[18] Etienne Coetzee, Bernd Krauskopf, and Mark H Lowenberg. The Dynamical Systems Toolbox: Integrating AUTO into MATLAB. In *16th US National Congress on Theoretical and Applied Mechanics*, State College, US-PA, 2010.

Appendix A: Aerodynamic Coefficients

Coefficients for the aerodynamic and thrust model of MAKO are given by Tab. 2.

Table 2: Coefficients for the MAKO aircraft.

$C_{\mathrm{L}0}$	-4.700×10^{-2}	–
$C_{\mathrm{L}\alpha}$	3.944	–
$C_{\mathrm{L}q}$	4.820	–
$C_{\mathrm{L}\eta}$	1.656×10^{-2}	–
$C_{\mathrm{D}0}$	2.313×10^{-2}	–
C_{DL}	1.897×10^{-1}	–
$C_{\mathrm{m}0}$	4.300×10^{-2}	–
$C_{\mathrm{m}\alpha}$	-3.234×10^{-1}	–
$C_{\mathrm{m}q}$	-1.683	–
$C_{\mathrm{m}\eta}$	-7.600×10^{-3}	–
$C_{\mathrm{F}0}$	1.342×10^{-1}	rev^{-2}
$C_{\mathrm{F}J}$	-1.975×10^{-1}	rev^{-1}
$C_{\mathrm{F}n}$	4.229×10^{-4}	$\mathrm{s\,rev}^{-3}$

Predictive Feedback Augmentation for Manual Control of a UAV with Latency

J. Cox*, KC Wong
The University of Sydney, Sydney, Australia

ABSTRACT

Teleoperation of unmanned aerial vehicles is hampered by communication delay, which causes feedback from command inputs to take considerable time to be displayed to the operator. For an international internet connection, round trip latencies can reach 500ms. The satellite connections used for military UAVs can have latencies on the order of seconds. This delay presents a substantial control problem which has been solved in the past by control abstraction (instead of "roll left" the aircraft might be instructed "go to these coordinates"). Manual control remains difficult. This study borrows the client-side prediction concept from multiplayer video games to attempt to address the control delay to allow manual control. An estimate of the change in the vehicle state due to the commands that are yet to affect the feedback is computed and then the feedback that the pilot receives is modified to reflect this predicted change. Because of this change, the pilot can see immediately the effect of the control inputs. This study has explored the concept and built a prototype system functional in real time for flight testing.

1 INTRODUCTION

Unmanned aerial vehicles (UAVs) are seeing widespread growth in many existing and new applications. Presently, UAVs are often operated within Line of Sight (LoS), where the control delays come primarily from the control radio system in use. Entry level hobby transmitters commonly achieve latencies on the order of 30ms. Advanced hobby radio systems boast ranges of 60km.

When operating via a local video link, camera latencies range from 40ms for the fastest analog systems to hundreds of milliseconds for digital systems.

Recently, hobby grade low latency digital transmission has been introduced, with latencies below 50ms achievable. Latency of wifi based transmission systems can exceed 100ms. Adoption has been low, likely due to cost and fragility.

*Email address: jeremy.cox@sydney.edu.au

Some military UAVs are operated from thousands of kilometres away via encrypted satellite connection, which imposes a round trip delay on the order of seconds. Often the terminal flight phases are controlled from ground crews near the runway. Due to the short time scales on which the state of the vehicle can change during takeoff and landing, the latency presents difficulty in control. Outside of terminal flight phases, the latency is acceptable as manouevres and disturbances occur on timescales much larger than the latency.

1.1 Internet Latency

The latency of packets sent over the internet varies depending on the source and destination. One way latency from Australia to the US is measured at around 150ms. Two way communications between an Australian and US location will double this to find the round trip time, as well as add latency from networking within each continent.

Control of a UAV over internet has been explored previously by the aerial international racing of unamnned systems (AIRUS) student project, of which the first author was a part. The impact of the latency on the control was found to be very significant, with even small latencies rendering a quadrotor UAV barely controllable. Previous work by team members has discussed the concept of a predictive aid and shown improvement in human control of a simulated system [1].

1.2 Teleoperation with Latency

In 1967, Ferrell and Sheridan [2] detailed for the first time the problem of teleoperation. Three main components of a remote operation system were described: a remote loop which acts to process tasks remotely (increasing the abstraction of the control); a supervisory loop which consists of the operator receiving information about the remote device's environment and specifies new commands; and a local loop which represents the operator's computer locally assisting (such as by modelling the remote system to present quasi-feedback). Increasing the command abstraction by allowing the remote loop to process the perceived environment and determine subtasks to achieve a higher level goal forms the majority of the systems proposed by Ferrell and Sheridan.

The operation of a robotic arm with delayed control is investigated by Bohren [3], who presents an assisted teleoperation architecture. A virtual robotic arm with a control delay is used to perform part of an assembly task. The visualisation of the scene is enhanced by displaying the user command overlayed on the robotic arm, as well as by estimating user intent. The user intent is estimated from the user inputs and the displayed scene that they are in response to, and then the user input is modified to try to effect the user intent in the real scene. With the assistance, test subjects showed substantial improvement in task completion time.

1.3 Latency in Multiplayer Videogames

Online videogames have previously undergone development similar to the proposed system. Bernier [4] details the progression of architectures, from a single server handling a number of clients that are entirely ignorant of the game mechanics to more modern configurations using client-side prediction and other more advanced routines. A number of the procedures detailed by Bernier are associated with foiling untrustworthy clients (cheaters) and it is noted that for military simulators, the clients are trusted.

The most basic feature discussed by Bernier is client-side prediction. In a dumb-client system with a network latency, the inputs from the client are sent to the server, which interprets how they change the game state, which is then sent back to the client. Movement of a player through an environment is given as a notable example where client-side prediction can be used. Instead of waiting for the server to acknowledge a movement command and update the game state, the client assumes that the command will be accepted by the server and processes the expected game state change by itself. The canonical version of the game state is handled by the server, so the client prediction may need to be retroactively altered. This may occur for example when the player input has an unexpected result due to another player's input that has not had time to be transmitted to the client or due to game state information that is hidden from the client. With client side prediction, the player perceives that their movement commands are being processed instantly, rather than with the round trip latency, and so their experience appears consistent regardless of latency in their connection.

Bernier claims that the success and longevity of a videogame requires a seamless multiplayer experience.

1.4 Flight Controller Modes

Quadrotor platforms are, in all practical cases, controlled via a digital control system. In these control systems, the operator commands particular flight variables, rather than controlling the motors and other actuators directly. The actuators used are the four (other numbers can be used but for simplicity only quadcopters are discussed) propellers, with the left/right, forward/back and left/right diagonal pairs each being used to generate roll, pitch and yaw moments respectively.

A review of common control schemes for quadrotors is presented in Appendix A:. Presently, the only widespread method of combating latency in UAV control is abstraction of the pilot's controls (move to a GPS coordinate instruction replacing roll/pitch/yaw commands). With UAV technologies expecting to see substantial growth in both civilian and military sectors, the controllability of UAVs at long range (and hence with control latency) is an important area of development. Current systems are either short range and low latency or long range and high latency. The tried and tested client side prediction concept from multiplayer videogames is a strong candidate for expanding the operability of UAVs at long range.

2 Proposed System Overview

To combat latency, the client-side prediction concept is applied to a UAV. The adaptation to a UAV can be summarised as predicting and displaying the vehicle state at the time when the commands being given currently will arrive at the vehicle. The eventual goal is to implement the scheme on a micro (112mm size) quadrotor flown in rate mode using a video feed from onboard as feedback. The operator receives information about the vehicle's state from the video feed, including the angular position from the image of the horizon.

Figure 1 shows the flow of commands and feedbacks around a time t_n at the remote station. 2τ is the roundtrip latency (the one-way latencies have been assumed to be identical and equal to τ).

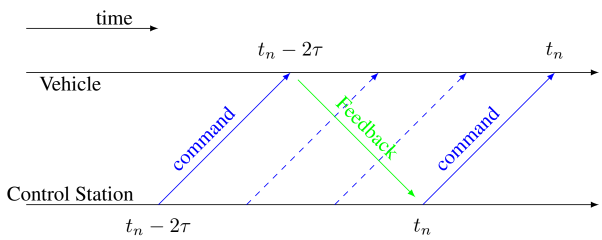


Figure 1: Flow of command and feedback.

The origin of the time axis is chosen slightly differently between the vehicle and the control station. $t = 0$ when the arm command is sent or received. This is convenient because a command occurs at the same time coordinate whether it is being sent or received.

At the control station at t_n, the remote station has the feedback that is delayed by τ, but it can also see the commands that have already been sent. The commands between $t_n - 2\tau$ and t_n will arrive at the quad after the feedback at $t_n - 2\tau$, and prior to the command being given at t_n and arriving at t_n. This command information is then used to predict the change in vehicle state, and hence the feedback that the operator would expect to see if there was no delay.

The mathematics of the problem can be described more simply as:

> Given the system state and command inputs to be executed, predict the system state after those command inputs have been applied.

which is a familiar problem from control theory with well developed means of solution.

Because the system model is likely to not exactly match the real system, and the delayed feedback is available, it is

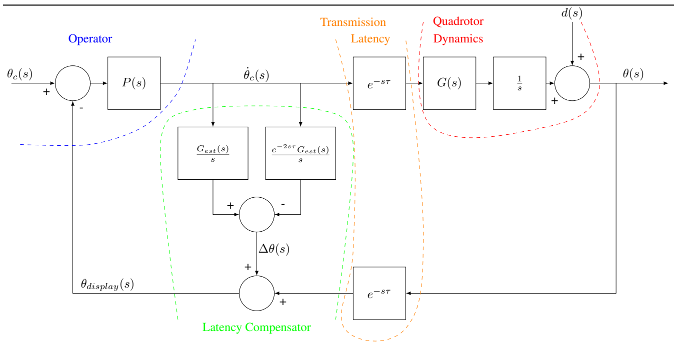


Figure 2: Block diagram with delay and compensator.

proposed to use the difference between two simulated vehicle states that are 2τ apart. One estimation is the state using all commands until time t_n and another only simulating until $t_n - 2\tau$. Taking the difference between these two states and adding the received feedback should give a more accurate estimate of the vehicle position than dead-reckoning simulation from start time to t_n. Errors in the model that accumulate prior to $t_n - 2\tau$ are contained in both terms, so are subtracted out.

2.1 System Block Diagram

Figure 2 shows the suggested system. The operator is modeled as determining a desired attitude, comparing that attitude to the feedback they are shown and then applying a controller to determine a command input. The compensator compares two simulated flight states, one of which is delayed by the round trip delay time and then augments the feedback by this amount. The remainder of the system is formed by the actual transmission delay and vehicle dynamics.

Because the implementation later in this paper deals with an angular rate controlled quadrotor, while the attitude feedback is an angular position from a video stream, some integrator blocks are included. These terms are shown for completeness and could be modified if a different control arrangement was used.

The diagram shown is for the one-dimensional case, where θ is any of the roll pitch or yaw axes. A real implementation will require propagation of these axes in three dimensions.

The compensator can be removed by setting $G_{est} = 0$, which gives the system with only the delay. The delayed system can be reduced to the typical system by setting $\tau = 0$, effectively removing the delay. Using the fact that the compensator and delay can be removed by substituting terms, the closed loop transfer functions need only be found for the delay and compensator case and the others will be readily found from those results.

2.1.1 Output and Reported Output Response to Command Derivation

First the closed loop transfer functions are found for the block diagram with the compensation system. For convenience, define

$$A(s) \equiv P(s)\left[G_{est}(s) + e^{-2s\tau}(G(s) - G_{est}(s))\right] \quad (1)$$

$$\approx P(s)G(s), \quad (2)$$

which approximately is equal to the latency free open loop transfer function for the pilot gain-quadrotor system if $G_{est}(s) \approx G(s)$. The displayed feedback and actual response can then be found:

$$\frac{\theta_{display}(s)}{\theta_c(s)} = \frac{A(s)}{s + A(s)}, \quad (3)$$

$$\frac{\theta(s)}{\theta_c(s)} = \frac{e^{-s\tau}P(s)G(s)}{s + A(s)}. \quad (4)$$

The delay term in the numerator is mandatory, as the input cannot reach the output without being subject to transmission delay (and a negative delay is infeasible without predicting $c(s)$ a short time in advance).

The disturbance response can also be found:

$$\frac{\theta(s)}{d(s)} = 1 - \frac{e^{-2s\tau}P(s)G(s)}{s + A(s)}, \quad (5)$$

which makes intuitive sense as the response is the disturbance plus a delayed, negative response to the disturbance. Once again the delay in the numerator is unavoidable as the disturbance must travel once around the loop to pass through the pilot and correct the disturbance.

2.1.2 Interpretation

The transfer functions for the compensated system can be used to find the delayed system and undelayed system for comparison. The control response transfer functions are shown in Table 1 and the disturbance response transfer functions in Table 2. With $A(s)$ highlighted in green and delay terms highlighted in red.

	$\frac{\theta(s)}{\theta_c(s)}$
No Delay	$\frac{P(s)G(s)}{s+P(s)G(s)}$
Delay	$\frac{e^{-s\tau}P(s)G(s)}{s+e^{-2s\tau}P(s)G(s)}$
Delay & Compensator	$\frac{e^{-s\tau}P(s)G(s)}{s+A(s)}$

Table 1: Transfer functions for the control response.

	$\frac{\theta(s)}{d(s)}$
No Delay	$1 - \frac{P(s)G(s)}{s + P(s)G(s)}$
Delay	$1 - \frac{e^{-2s\tau}P(s)G(s)}{s + e^{-2s\tau}P(s)G(s)}$
Delay & Compensator	$1 - \frac{e^{-2s\tau}P(s)G(s)}{s + A(s)}$

Table 2: Transfer functions for the disturbance response.

The introduction of the delay causes a delay term in the numerator of the command response and the counteraction to the disturbance.

The delay term in the numerator is caused by the forward latency for the command response and by disturbances needing to be observed and then responded to for the disturbance response. Addressing these delay terms is beyond the scope of this paper.

A delay term is also seen in the denominator. This term is caused by the operator being unable to immediately see the feedback from their inputs.

The introduction of the delay compensator modifies the term in the denominator. The term $A(s)$, highlighted in green, still contains the delay term as the feedback from the vehicle is incorporated. If a suitably accurate estimate of the vehicle dynamics $G_{est}(s)$ is known, then the term in the denominator becomes approximately equal to the undelayed denominator. The compensator acts to provide feedback that shows immediately the effect of the command inputs, removing the effect of the delay from the denominator.

The change that the compensator brings to the denominator makes the transfer functions much more similar to the no delay case. The command response is affected only by the forward latency, which delays the response but otherwise does not affect its behaviour. The disturbance response is improved by the compensator, but it will still take at least the roundtrip latency before the pilot begins to correct disturbances.

2.2 *Estimation Error*

If the vehicle attitude is estimated by integrating a model which provides the vehicle attitude rate, then

$$\theta_{est}(t) = \int_0^t \dot{\theta}_{est} dt \tag{6}$$

$$= \int_0^t \dot{\theta} + \dot{\theta}_{est.error} dt \tag{7}$$

$$= \theta(t) + \int_0^t + \dot{\theta}_{est.error} dt. \tag{8}$$

Where θ represents an angular position, $\dot{\theta}_{est}$ is an estimate of the angular rate obtained from a model of the dynamics applied to the recorded inputs and $\dot{\theta}_{est.error}$ is the error in $\dot{\theta}_{est}$. The issue with a dead reckoning approach can be seen here; the size of the error will tend to grow with flight time. If the displayed attitude is based on an out of date attitude plus a prediction of its upcoming changes, then

$$\theta_{display} = \theta_{est}(t_n) - \theta_{est}(t_n - 2\tau) + \theta(t_n - 2\tau) \tag{9}$$

$$\begin{aligned} &= \theta(t_n) + \int_0^{t_n} + \dot{\theta}_{est.error} dt \\ &\quad - \left[\theta(t_n - 2\tau) + \int_0^{t_n - 2\tau} + \dot{\theta}_{est.error} dt \right] \\ &\quad + \theta(t_n - 2\tau) \end{aligned} \tag{10}$$

$$= \theta(t_n) + \int_{t_n - 2\tau}^{t_n} \dot{\theta}_{est.error} dt \tag{11}$$

Only errors in the model that manifest between $t_n - 2\tau$ and t_n form part of the displayed system state. Importantly, the error does not accumulate with time and large errors (such as from disturbances or rapid motions where model errors are amplified) will only manifest for the delay time. Also, so long as the latency 2τ is small, the error in the displayed information is minimal. If a dead reckoning approach is used, then the error will increase by roughly a factor of $t_n/2\tau$, and will hence grow with time.

3 System Identification

3.1 *Introduction*

To predict the motion of the vehicle, a dynamic model is needed.

For the quadrotor vehicles considered in this paper, the dynamics of the three rotational axes were assumed to be independent of each other and the flight condition (speed, altitude, throttle setting). This assumption is valid only in rate mode.

The development of high rate flight data recording [5] allows system identification by measurement rather than modelling. Quadrotor models do exist, but model measurement was chosen due to the importance of accuracy to the specific hardware and software configuration. A large number of vehicle geometry and parameters would also be needed for a derived model approach, many of which are difficult to measure or have errors in the manufacturer's specification.

3.2 *Obtaining a Model*

Finding a model of each axis from the recorded input and output data was achieved using the inbuilt `tfest` function in MATLAB. This function accepts one or more logs of system input/output data and returns a transfer function that closely maps the input data to the output data. The function works by creating an initial guess and then iterating to minimise the error in the transfer function.

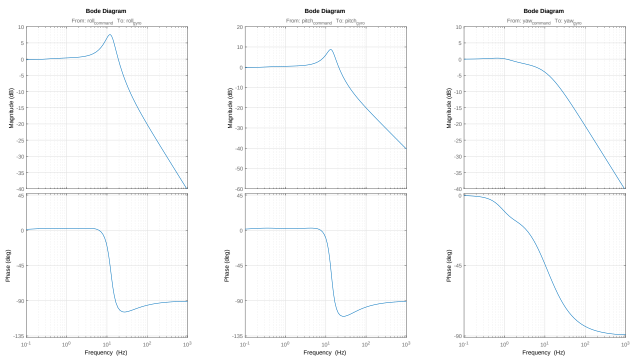


Figure 3: The models generated for the 112mm quadrotor.

3.3 Identified System

For the 112mm quadrotor used, eight flight logs were used to generate a system model. The procedure was validated by generating eight models with a log each excluded and verifying that the models remained valid on the excluded log. Bode plots of the system model generated from all eight logs are shown in Fig. 3. The models map angular rate command to angular rate.

4 Augmenting the Feedback

With the change in vehicle attitude predicted, the feedback can be augmented to reflect this prediction. The video is first undistorted to remove fisheye effects from the camera lens, and then warped so that the horizon (and other objects far from the camera) appear to move immediately with operator input.

Figure 4 shows an example of image warping for a prediction of rolling. In this example the feedback shows the quadrotor in a near inverted attitude, but the command inputs that are yet to arrive will bring the quadrotor to nearly level, so the compensator warps the image such that the horizon appears level. Warping for the yaw and pitch axes is achieved by shifting the image left and right, with the camera field of view determining the rate (pixels per degree) of shifting.

5 Simulated Predictive Performance

Before implementing the compensator, it is desirable to estimate the predictive performance and to understand how the duration of the latency will affect the performance. The predicted position can be expressed in terms of the actual position and the error in the modeled system as

$$\theta_{display} = \theta(t_n) + \int_{t_n-2\tau}^{t_n} \dot{\theta}_{est.error} dt, \qquad (12)$$

from equation 11. For efficient calculation the error term is found in terms of the angular position error accumulated from

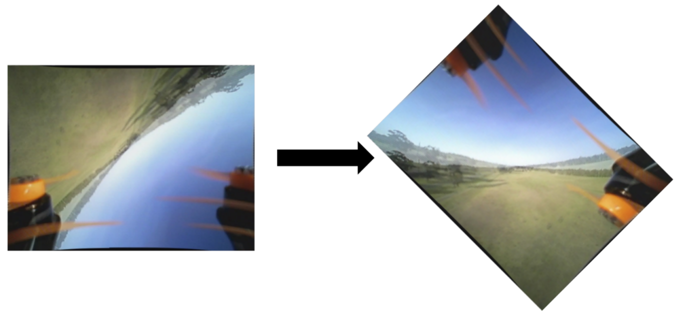


Figure 4: Example of image warping. The image is warped to reflect the compensator predicting a 135 degree roll to the right.

the beginning of the flight

$$\int_{t_n-2\tau}^{t_n} \dot{\theta}_{est.error} dt = \int_0^{t_n} \dot{\theta}_{est.error} dt - \int_0^{t_n-2\tau} \dot{\theta}_{est.error} dt. \qquad (13)$$

The accumulated error from the start of the flight is found by integrating the difference of the measured output and the output modeled from the measured input. The difference across a moving window is then taken to find the prediction errors throughout a flight log. The predictive performance was only measured on logs not used to generate the model used to assess performance (as a model derived from the flight log after a flight cannot be used to predict during that flight).

A histogram of the prediction errors was computed to visualise the performance. A histogram of the typical prediction errors on the roll axis with a 1 second delay is shown in figure 5. The standard deviation of predictive errors with a 1 second delay for this flight is 1.1^o, and most flights show similar distributions of errors, with no more than a couple of degrees standard deviation. A more thorough analysis of the simulated performance, and variation in performance with latency can be found in [6]. With predictive performance within a few degrees, the concept is now ready for implementation and proof of concept.

6 Implementation

The latency compensation system was implemented on the 112mm size quadrotor. The quadrotor was fitted with a fixed camera with analog transmission. The real time implementation had three key parts:

- Introduce an artificial latency
- Measure the command inputs and generate the prediction of the feedback's change
- Receive and augment the video stream

The Node.js environment was chosen for implementation. Node.js is a javascript runtime, designed for internet connected applications and has event based features. The event

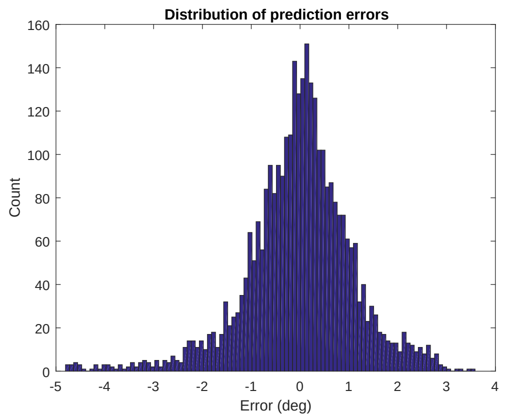


Figure 5: Histogram of predictive errors on the roll axis throughout a flight, for a delay time of 1 second.

based features are useful for running the simulation in real time, I/O in real time and introducing delays. There is a large ecosystem of pre-existing libraries for a variety of tasks, including mathematical tasks, image processing and I/O. The Node.js environment will also be ideal for future work implementing on a real source of latency. Node.js is fast, but being an interpreted javascript environment it is not fast enough to run the image warping in real time.

OpenCV was used to perform the image undistortion and warping, which was faster than Node.js because it runs as precompiled C code.

The architecture for testing the compensator is shown in Fig. 6. Regular flight uses Node.js as a passthrough to display video, to give a baseline. Flying with latency uses the Node.js environment to introduce latency to the system, and latency compensated flight uses Node.js to introduce latency and augment the feedback. All command and feedback passes through the Node.js environment.

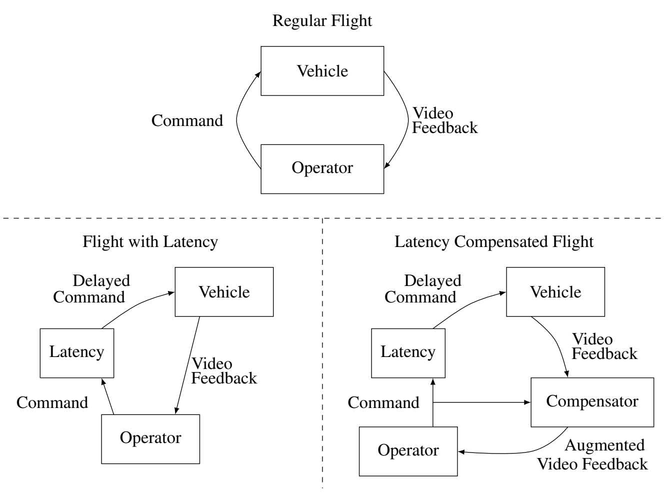


Figure 6: The flight test architectures.

Figure 7 shows the simulation arrangement used. The simulation is indicated in blue, beginning from the start of the flight. The vehicle states in red are simulated and still in memory at t_n. The simulated vehicle states at t_n and $t_n - 2\tau$ are compared to find the expected change in the feedback. A single propagation of the flight was used as it is difficult to guarantee that two simulations running side by side but separated in time will run with identical inputs, and hence may be prone to drift apart. In future work, input to the simulation from telemetry feedback may be necessary to ensure that the simulation does not drift from the true vehicle state. Because the quadrotor's angular response does not vary with flight condition, this architecture is sufficient.

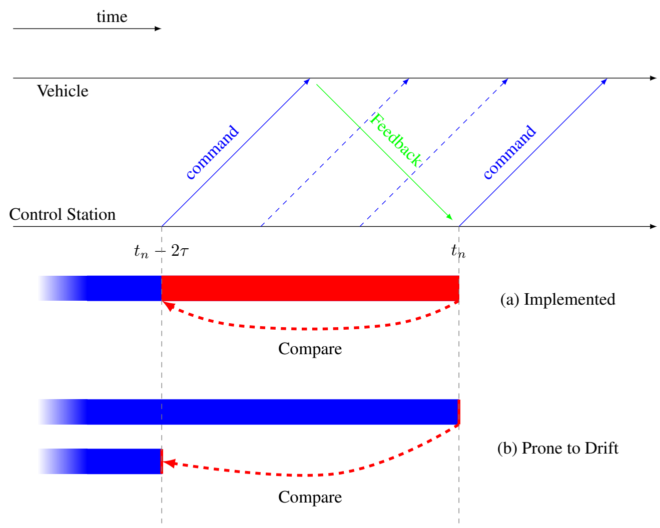


Figure 7: Simulation and computation of the predicted feedback.

7 Flight Testing and Results

The 112mm quadrotor was flown without any latency, with latency and with compensated latency.

The test pilot did not have any remarks about abnormal flight behaviors when flown without latency. The size of the quadrotor meant that its position was quite susceptible to gusts.

Latencies of 500ms and 1000ms were tested. With a 500ms latency, the test pilot struggled to maintain control. Early flights resulted in crashes. Later flights the test pilot was able to maintain flight, but only achieving the most basic maneuvers and tasks. Control was achieved by giving brief inputs and then waiting to observe the effect of those inputs before determining the next input. Controlled flight was not achieved with a 1000ms delay.

Compensated latencies of 500ms and 1000ms were tested. The test pilot remarked immediately when flying with 500ms of compensated latency that the aid allowed smooth control of the vehicle. With both 500 and 1000ms of compensated latency, tasks such as following a fence line were achieved. Controlled flips in the roll and pitch axis were achieved. Figure 8 shows the pilot's view at various points through a flip at 1000ms latency. The flips highlighted that not accounting for the delays in the various command and feedback interfaces had an effect on the prediction. The compensator expected the video to show the vehicle rolling briefly before it did, so the horizon dipped from level as the quad was "catching up" to the pilot. This bug will be addressed in future flight testing.

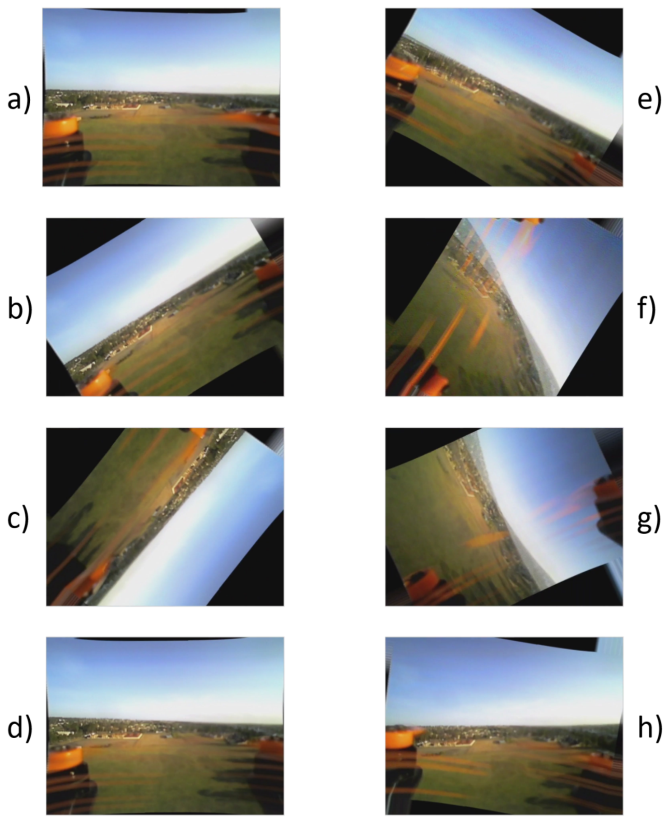


Figure 8: Frames displayed by the compensator during a flip flown with 1000ms latency. The issue introduced by the latency not accounted for can be seen in frames f) and g). Frames are labeled a)-h) chronologically. Frames on the left are taken from when the flip is commanded (vehicle remains level) while frames on the right are taken from when the vehicle executes the flip (horizon should appear to remain level, but doesn't because the compensated latency is slightly less than the true latency). The total time between the first and last frame is approximately 2 seconds.

8 Conclusion

The problem of teleoperation has been explored, and the client side prediction model has been borrowed from multiplayer videogames to address the control latency problem in UAVs. The mathematical basis for the concepts viability has been explored and a scheme that does not suffer from drift has been found. The concept has been implemented and proof of concept achieved in flight testing. Future work may refine the system, explore systems where drift of intermediate states like airspeed could affect the system response or explore implementation with a real latency source (such as an internet connection).

References

[1] Tara Bartlett. Delay compensation for international unmanned aerial vehicle control "unpublished". *AERO3711 Final Report*, 2017.

[2] W.R. Ferrell and T.B. Sheridan. Supervisory control of remote manipulation. *pp 16-17 of NEREM Record. Northeast Electronics Research and Engineering Meeting, Boston, November 2–4, 1966. Volume VIII. Newton, Mass., Ernest E. Witschi, Jr., 1966.*, Oct 1967.

[3] Jonathan Bohren, Chris Paxton, Ryan Howarth, Gregory D. Hager, and Louis L. Whitcomb. Semi-autonomous telerobotic assembly over high-latency networks. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, HRI '16, pages 149–156, Piscataway, NJ, USA, 2016. IEEE Press.

[4] Yahn Bernier. Latency compensating methods in client/server in-game protocol design and optimization. *Game Developers Conference*, 98033(425), 2001.

[5] Boris B. Betaflight. `https://github.com/betaflight/betaflight`, 2013-2018.

[6] Jeremy Cox. Adapting uav control for latency "unpublished". *Undergraduate Thesis*, 2017.

[7] GitHub Contributors. ArduPilot. `https://github.com/ArduPilot/ardupilot`, 2011-2018.

[8] GitHub Contributors. PX4 Pro Drone Autopilot. `https://github.com/PX4/Firmware`, 2011-2018.

Appendix A: Quadrotor Control Modes

The following review of control modes is based on material from the ArduPilot [7], PX4 [8] and BetaFlight [5] open source flight control codes and their associated user guides.

A.1 Basic Control Modes

The most basic control systems use the attitude or attitude rates as the process variables, as well as an overall throttle setting (which is not subject to a closed loop control system). The three angular inputs typically use a self-centring gimbal for input, and an axis which holds its position for throttle.

The inexpensive 'toy' grade models typically implement these control schemes. These control systems require only a gyro, as well as an accelerometer for attitude control modes, which are easily and inexpensively surface mounted aboard a flight computer board.

A.2 Rate Mode

Also referred to as: acro mode.

The most basic controllers use a gyro only to control the attitude rates, requiring three axes of input for the attitude rates and one for the throttle setting. The relative thrust setting of the opposite pairs of motors is used as the actuator for each axis. The throttle input determines the overall sum of all of the motor throttle settings. A basic block diagram for this arrangement is shown in Fig. 9. While this mode is quite basic and simple to understand, it is one of the trickiest to fly (though it allows the largest set of possible maneuvers and most direct control) due to its lack of self-leveling.

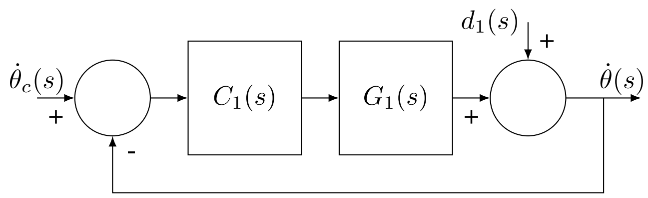


Figure 9: Rate mode controller typical block diagram.

A.3 Angle Mode

Also referred to as: stabilise, self-level mode.

Angle mode is fairly similar to rate mode, except that instead of controlling the angular rates, the pitch and roll angular position is determined by the stick inputs. The stick inputs are mapped to angular positions such that a zero input puts the quadcopter in a level, hovering attitude. Pitch/roll inputs are used to place the vehicle at an angle, which accelerates the quadcopter horizontally. This control is achieved using a new control loop, with the rate control loop from the rate mode as the plant, shown in Fig. 10.

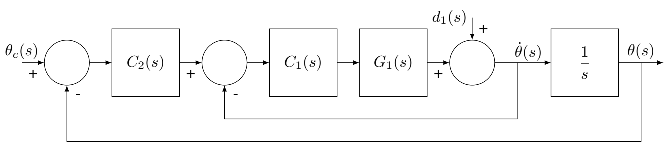


Figure 10: Angle mode controller typical block diagram.

The yaw axis remains an angular rate controller rather than angular position controller (so that a zero stick input gives a steady heading, rather than moving the quadcopter to a particular heading angle).

A.4 Advanced Control Modes

More advanced systems build on the basic control modes, providing as setpoints to basic control modes inputs based on other sensors, so that setpoints may be set for other variables. Commonly available sensors include:

- barometer, which can be inexpensively surface mounted on a flight computer board
- GPS, which is often an external module connected to the flight computer via a serial connection
- optical flow, which typically requires a substantial processor alongside the flight computer
- ultrasonic sensor, which allows sensing proximity to obstacles in a particular direction

With these additional sensors, the vehicle position and position rate (on some or all axes, depending on sensor) can be used as a process variable. This allows flight modes such as position or altitude hold. Other control modes allow preprogrammed setpoints to be used, such as orbiting a point at a constant rate or following waypoints.

Abstraction of operator control means changing the operator's control from things like "roll left" to "move to these coordinates" and can be an effective means of combating latency.

Design, Fabrication, and Flight Test of Articulated Ornithopter

Seongyoung Kim[1,*], Myungkang Kim[1,†], Seungkeun Kim[1,‡] and Jinyoung Suk[1,§]
[1]Department of Aerospace Engineering, Chungnam National University, Daejeon, 34134, Republic of Korea

ABSTRACT

Recently, research on flapping wing aircraft is on the rise due to the weight reduction of sensors and actuators. However, studies on articulated ornithopter have been lacking and commercial articulated ornithopter is hard to find. In this paper, we design and fabricate an articulated ornithopter with flapping frequency of 2-3Hz and span of 1.8m. The design based on kinematic analysis is verified through Matlab and Solidworks, and Adams. The platform is made of carbon plate with EPP material skin. The design parameters are compared and verified using a motion capture camera. Additionally, this paper shows thrust analysis with respect to wing shapes sweptback and rectangular. Finally, the design parameters are verified and analyzed through a motion capture camera.

1 INTRODUCTION

The bird is an efficient and superior flying object with over 150 million years of evolution. Humans have longed to fly in the sky watching these birds. Leonardo da Vinci (1452 1519) first made wing flap wings, and in 1924 the mechanism for flapping wing aircraft was studied.[1] In 1930 Lippisch's early work was carried out and many attempts were made to imitate the flight of birds in a technical approach.[2] In the 1980s, the energy benefits of airflow with winged wings were studied.[3]
Recently, due to the weight reduction of the mounted equipment, interest in the winged flight robot is increasing. In 2015-2017, Chungnam national university, Korea, has carried out on system identification, route point flight, etc. using commercial winged robots of single articulated robots.[4] However, since a single articulated robot has a short span length, it requires a wing flap of 7 Hz or more in order to generate thrust and lift and is not suitable for energy saving effect.
On the other hand, the composite articulated winged robot can generate thrust and lift at the wing of 2-3 Hz because of its long span length. Also, real birds generate positive aerodynamics at downstroke and negative aerodynamics at upstroke. In the upstroke, the wing is folded to reduce the resistance and reduce the aerodynamic drag. By reducing inertia moment, the efficiency of flight can be increased.[5][6][7]
Smartbird, a complex articulated winging robot based on the shape of a gull, was developed by Festo in Germany in 2011. Smart bird is equipped with a servo on a wing tip to obtain a positive aerodynamic force by attaching a Hall sensor to the gear and using carbon plate and extruded polyurethane foam for weight saving. The specification of Smartbird is shown in Table 1.[8]

Figure 1: Smartbird

Span	2m
Weight	500g
Flapping Frequency	2-3Hz
Flight speed	4.7m/s

Table 1: Smartbird specification

Inspired by Smartbird, a few universities have been working on a complex articulated winging robot. In 2016, the Chinese graduate school of Harbin Institute of Technology conducted a composite articulated wing flapping kinematics study and analyzed it to make a flying body.[9] However, they did not analyze the design parameter and mechanism of the articulated winging robot using motion capture camera. In 2014, King Abdullah University of science and technology of Saudi Arabia conducted experiments on thrust and lift according to the wing shape. In this study, it was verified that the sweptback wing shape is larger in thrust and lift than the straight wing shape. However, this is the result of the UVLM simulation, which is not applied to the actual winging robot.[10]
In this paper, Chapter 2 describes an articulated winging robot is designed and fabricated through kinematic analysis. chapter 3 analyzes the design parameters are verified and analyzed

*Email address(es): ks8600@hanmail.net
†Email address(es): kmkaero3746@gmail.com
‡Email address(es): skim78@cnu.ac.kr, corresponding author
§Email address(es): jsuk@cnu.ac.kr

through the motion capture camera. Chapter 4 analyzes the thrust according to the wing shape and area with a single axis load cell. Chapter 5 analyzes the flight test results.

2 Design and Fabrication of Ariticulated Ornithopter

This paper, designs and fabricats a 1.8m - class articulated wing-like body with a flap frequency of 2-3Hz based on kinematic equations. We verified it through Matlab and Solidworks, a 3D modeling tool, and Adams, a multibody dynamics simulation. The robot frame was made of carbon plate, and the skin was made of EPP material.[9]

2.1 *Drive Mechanism Design*

The articulated wing mechanism used in this paper was motivated by the Smartbird mechanism [8].
The power starts with a brushless motor and is transmitted to the main gear via the reduction gear. In order to operate in the frequency range of 2-3 Hz, the gear reduction ratio is designed as 44, which reduces the load on the motor. Main gear is connected to crank, coupler, and rocker (four-bar linkage) which transmit power to upper-spar and lower-spar, respectively.

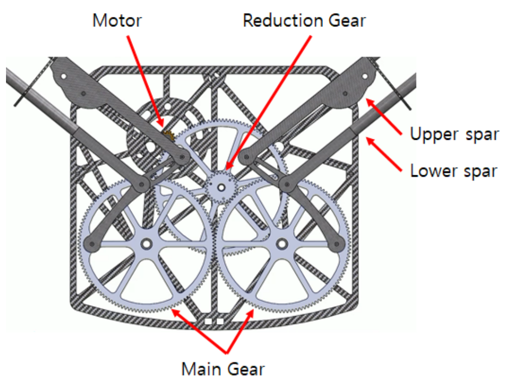


Figure 2: Drive mechanism

	Main Gear	Reduction Gear	Motor Gear
Teeth	120	27	12
Module	0.6	0.6	0.6
Diameter (mm)	73.2	17.4	8.4
Width (mm)	5	8	5

Table 2: Gear information

2.2 *Main wing Design*

The airfoil was modified in the same NACA7412 as the Smart bird. The wing mechanism of the articulated winged robot is similar to the human arm [11]. It is divided into three parts : the shoulder joint, the elbow joint and the wrist joint. The shoulder joint is divided into an upper-spar and a lower-spar. The upper-spar generates flapping motion, and the lower-spar produces translational motion. This motion of the shoulder joint is transferred to the elbow joint.
An elbow joint causes folding of the outer wing.[12] In order to generate thrust and lift positively, the wrist joint is in the upstroke state and the airfoil of the outer wing is in the pitch up state. In the downstroke, the airfoil of the outer wing is in the pitch down state.[5]
Considering this point, bearing is mounted on the wrist joint so that the twist angle of the wing is formed.
In 2012, DGIST conducted research on flapping-wing model for aerial robot. Through this study, the articulated winging robot can obtain the ideal aerodynamic force when the length ratio between the inner wing and the outer wing is 1:2.[13] Therefore, the inner wing length was designed to be 30 cm and the outer wing length to 60 cm.

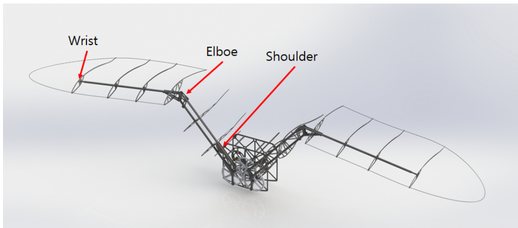


Figure 3: Wing mechanism

2.3 *Kinematic analysis*

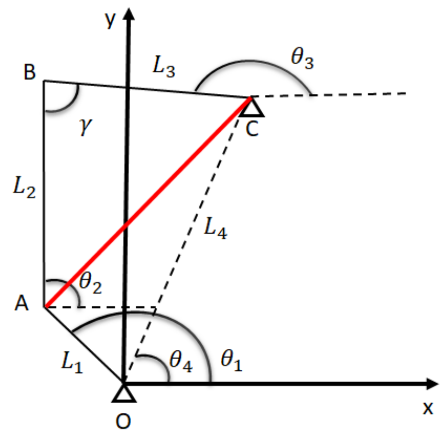


Figure 4: A schematic of the inner flapping wing mechanism

In this paper, the name of the articulated flapping robot designed and manufactured is USGull, and Figure 2 shows the wing mechanism of the flapping of the USGull. Fig. 2, the mechanism of the USGull is a four-bar link with a Crank-Rocker structure, named as follows. (L_1=Crank, L_2= Coupler, L_3= Rocker, L_4= Ground)
The derivation of transmission angle and the inner flapping angle is as follows.

$$\overline{AC} = \sqrt{L_1^2 + L_4^2 - 2L_1L_4cos(\theta_1 - \theta_4)} \quad (1)$$

$$\gamma = cos^{-1}\left(\frac{L_2^2 + L_3^2 - \overline{AC}^2}{2L_2L_3}\right) \quad (2)$$

$$\theta_3 = 2tan^{-1}\left(\frac{L_1sin(\theta_1 - \theta_4) - L_2sin(\gamma)}{L_1cos(\theta_1 - \theta_4) + L_3 - L_4 - L_2cos(\gamma)}\right) \quad (3)$$

γ is the transmission angle and should be within the range of 45°-135° because the four-bar link design needs satisfy the design conditions[14]. The larger the flapping angle and the span, the greater the thrust becomes.[15] In this paper, the span length is 1.8m, Θ_3 is the Inner flapping angle, the design condition of this paper is set to $L_1 = 29mm$, $L_2 = 65.2mm$, $L_3 = 63mm$, $L_4 = 86.9mm$. θ_4 = 67° which is the input value.

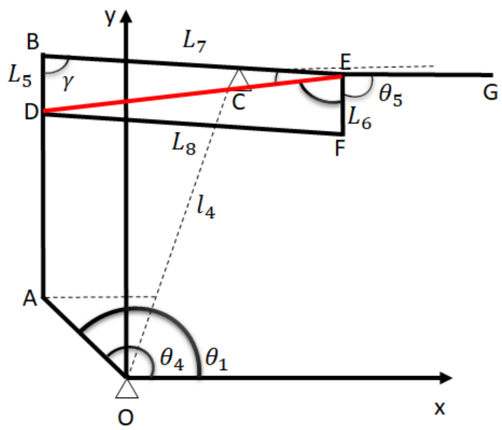


Figure 5: A schematic of mechanism in kinematics

For the outer wing mechanism (as shown in Fig. 5), to achieve the good performance, the quadrilateral mechanism (BDFE) should be a parallelogram. Thus, the folding angle of the inner and outer wing is writhen as eq.7

$$\overline{DE} = \sqrt{L_7^2 + L_5^2 - 2L_5L_7cos(\gamma)} \quad (4)$$

$$\angle BED = cos^{-1}\left(\frac{\overline{DE}^2 + L_7^2 - L_5^2}{2\overline{DE}L_7}\right) \quad (5)$$

$$\angle FED = cos^{-1}\left(\frac{\overline{DE}^2 + L_8^2 - L_6^2}{2\overline{DE}L_7}\right) \quad (6)$$

$$\angle Folding = \angle BED + \angle FED + \theta_5 \quad (7)$$

$\overline{DE}$ is a function of γ by Eq.(4) and γ is a function of $\overline{AC}$ by Eq.(2) and $\overline{AC}$ is a function of the input value by Eq.(1). In this paper, we set $L_5 = 25.5mm$, $L_6 = 25.5mm$, $L_7 = 249mm$, $L_8 = 249mm$ and θ_5 = 71°.

2.4 *Simulation analysis*

This paper verified the results by comparing the results of MATLAB with those of ADAMS in terms of transmission angle, inner flapping angle, folding angle. Input (Θ_1) is excited, and the result is shown in Figure 6-9.

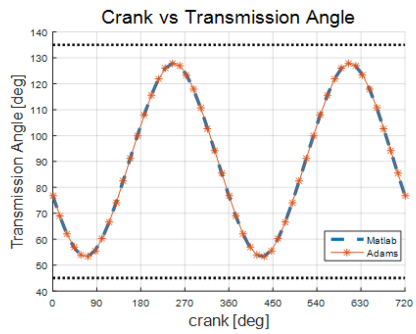


Figure 6: Crank vs Transmission algle

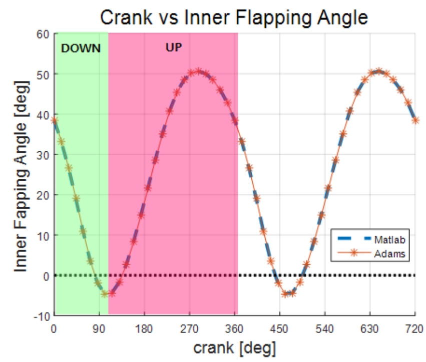


Figure 7: Crank vs Inner flapping angle

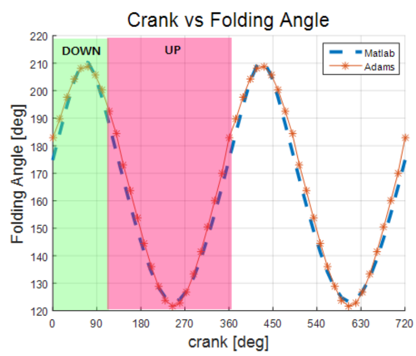


Figure 8: Crank vs Folding angle

2.5 *USGull Prototype*

In order to meet weight and durability, the frame was made with carbon plate, and EPP material was used as the skin. Figure 10 and Table 3 show the prototype and detailed specifications of the USGull respectively.

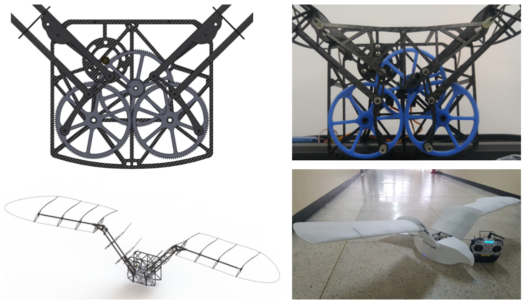


Figure 9: USGull prototype

Weght	460g	Mean chord	0.25m
Span	1.8m	Aspect ratio	6.8
Length	0.9m	Skin	EPP
Gear ratio	1:44	Flapping Frequency	2-3Hz

Table 3: USGull specification

3 Analysis with motion capture camera

As shown in Fig.10, the experimental environment of the motion capture camera was constructed and the kinematic design parameters were verified through this experiment.

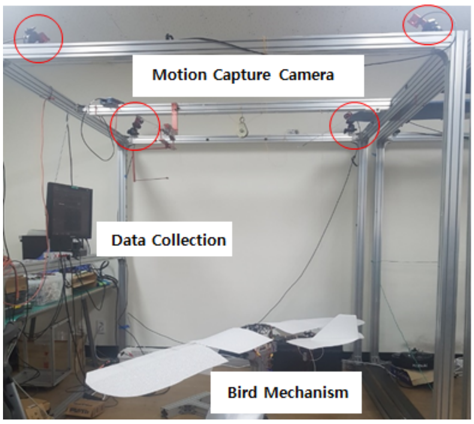


Figure 10: Experiment Environment

Fig.11-12 show the results of the analysis using the motion capture camera. It can be confirmed that the design parameters are designed so as to be equal to each other when compared with the kinematic equations and the ADAMS simulation obtained in Fig.7-8.

4 Thrust test with Load Balance

Thrust is an important factor when an aircraft takes off. This is true of birds flying. The research was conducted on the

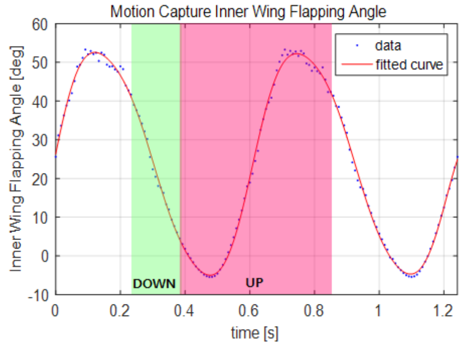


Figure 11: Motion Capture Inner Wing Flapping Angle

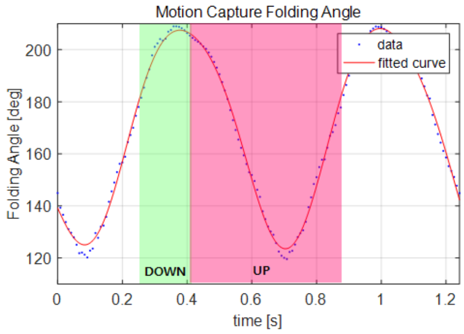


Figure 12: Motion Capture Folding Angle

optimal wing shape that can achieve maximum efficiency by utilizing UVLM simulation.[16] The conclusion of the study was that thrust was higher in the sweptback wing than in the straight wing.

In this paper, the thrust due to the wing movement, rather than the wind tunnel test environment, was carried out according to the wing area and wing shape. For the experiment, a one-axis load cell was used and the value of the change in thrust according to the angle was measured. The experimental environment is shown in Fig.13.

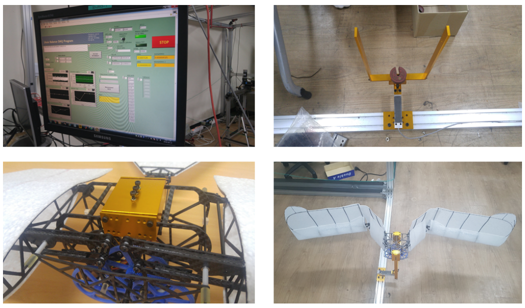


Figure 13: Experiment enviroment

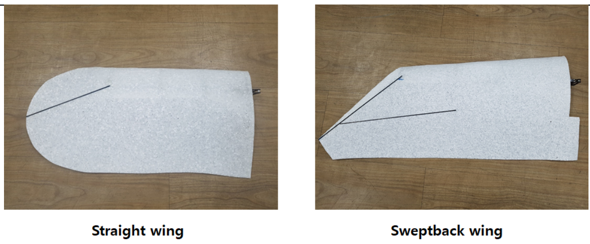


Figure 14: Wing shape

4.1 *Sweptback wing vs rectangular wing*

As shown in Fig.14, the comparison was made between straight and sweptback wings when the areas were the same ($0.1352m^2$).

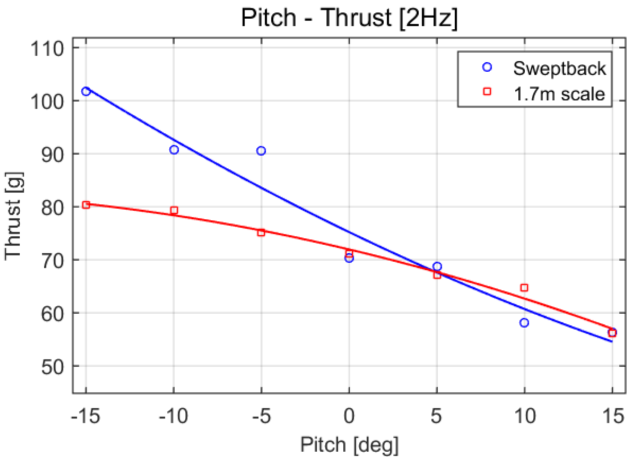


Figure 15: Thrust result of different wing shape

Fig.15 shows the results of the thrust test. Overall, the thrust shows a maximum at -15°. This is because the thrust due to the wing is higher than the center of gravity. Also, it was confirmed that when the pitch angle is negative with respect to the pitch angle of 0°, the thrust of the sweptback wing is increased by about 20% than the rectangular wing.

4.2 *Comparison of rectangular wings with different wing areas*

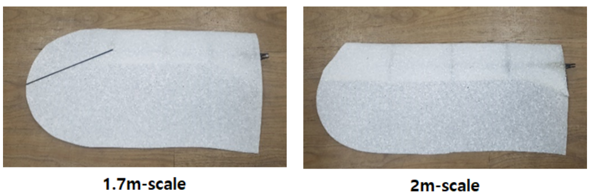


Figure 16: Same shape with defferent area

Fig.16 shows a straight wedge with the different area. The area of 1.7m-scale is $0.1352m^2$, and the area of 2m-scale is $0.1716m^2$. The span length is 2m-scale longer than 1.7m-scale and about 15cm longer.

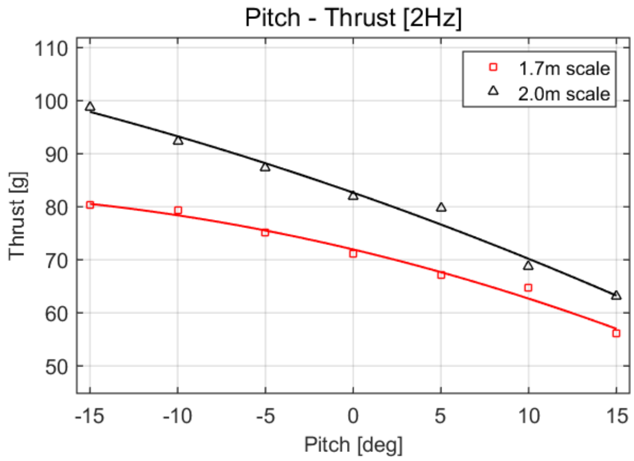


Figure 17: Thrust result of different wing area

Fig.17 shows the results of the thrust test. As shown in the graph of Fig.15, the maximum thrust is found at -15°. Also, the larger the area, the greater the overall thrust. However, as the wing area increases, the load on the elbow joint increases and the flapping mechanism becomes unstable. Therefore, it is necessary to adopt a double elbow joint to make a structural complement.

5 Preliminary Flight Test Results

Figure 18: Flight Test

We conducted a preliminary flight test and performed a performance test on the USGull with 1.7-scale. It is believed that maneuverability and stability are secured. However, due to the periodical wing movement, there was a structural problem of the elbow joint and the flight performance was not as good as desired.

6 Conclusions and Future work

In this paper, an articulated ornithopter was designed and fabricated through kinematic modification and verification.

kinematic parameter design verification was performed using a motion capture camera. Also, the thrust test was performed according to the wing shape and area, and the results were compared and analyzed. Finally, the stability and maneuverability of the ornithopter were analyzed through the flight test, but due to the structural problems on the elbow joint, the continuous flapping movement was not performed.
After the elbow joint is structurally reinforced, the flight test will be conducted and the system identification will be carried out on the articulated ornithopter using the flight data.

Acknowledgements

This research was supported by a grant to Bio-Mimetic Robot Research Center funded by Agency for Defense Development and Defense Acquisition Program Administration. (UD130070ID) and the research project (10062327) funded by the Ministry of Trade, Industry and Energy of Korea, Republic of.

References

[1] Alexander M Lippisch. Man powered flight in 1929. *The Aeronautical Journal*, 64(595):395–398, 1960.

[2] Walter Birnbaum. Das ebene problem des schlagenden flügels. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 4(4):277–292, 1924.

[3] William McKinney and James DeLaurier. Wingmill: an oscillating-wing windmill. *Journal of energy*, 5(2):109–115, 1981.

[4] Jangjin Oh, Seongyoung Kim, Byoungju Lee, Seungkeun Kim, and Jinyoung Suk. System identification & attitude control of avian-type flyer with flight test. In *Control Conference (ASCC), 2017 11th Asian*, pages 1677–1682. IEEE, 2017.

[5] Irby J Lovette and John W Fitzpatrick. *Handbook of bird biology*. John Wiley & Sons, 2016.

[6] Ismail H Tuncer and Mustafa Kaya. Optimization of flapping airfoils for maximum thrust and propulsive efficiency. *AIAA journal*, 43(11):2329–2336, 2005.

[7] James D DeLaurier. An aerodynamic model for flapping-wing flight. *The Aeronautical Journal*, 97(964):125–130, 1993.

[8] Wolfgang Send, Markus Fischer, Kristof Jebens, Rainer Mugrauer, Agalya Nagarathinam, and Felix Scharstein. Artificial hinged-wing bird with active torsion and partially linear kinematics. In *Proceeding of 28th Congress of the International Council of the Aeronautical Sciences*, 2012.

[9] Hongli Jiang, Chaoying Zhou, and Peng Xie. Design and kinematic analysis of seagull inspired flapping wing robot. In *Information and Automation (ICIA), 2016 IEEE International Conference on*, pages 1382–1386. IEEE, 2016.

[10] Mehdi Ghommem, Nathan Collier, Antti H Niemi, and Victor M Calo. On the shape optimization of flapping wings and their performance analysis. *Aerospace Science and Technology*, 32(1):274–292, 2014.

[11] Taku Yokoyama, Kazuo Tanaka, and Hiroshi Ohtake. Development of a variable-wing mechanism based on flapping motion of birds. In *SICE Annual Conference, 2008*, pages 168–173. IEEE, 2008.

[12] Nandu Jith PJ and Harsh Gupta. Design and fabrication of a flapping wing unmanned aerial vehicle with bird kinematics. *Journal of Aerospace Engineering & Technology*, 4(2):9–21, 2014.

[13] Youn-Ho Choi, Jung-Eun Joung, and Dong-Ha Lee. Flapping-wing model for aerial robot. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2012 9th International Conference on*, pages 180–181. IEEE, 2012.

[14] David H Myszka. *Machines and mechanisms*. Prentice Hall, 2004.

[15] Lung-Wen Tsai and Sameer Joshi. Kinematic analysis of 3-dof position mechanisms for use in hybrid kinematic machines. *Journal of Mechanical Design*, 124(2):245–253, 2002.

[16] Mehdi Ghommem, Nathan Collier, Antti H Niemi, and Victor M Calo. On the shape optimization of flapping wings and their performance analysis. *Aerospace Science and Technology*, 32(1):274–292, 2014.

Wing Sweeping Mechanism for Active Control and Stabilisation of a Flapping Wing MAV

Diana Olejnik *, Aadithya Sujit †, Matej Karasek, Bart Remes, Guido de Croon
MAVlab, Control and Operations, Faculty of Aerospace Engineering, TU Delft

ABSTRACT

During flight, natural fliers flap, twist and bend their wings to enhance flight performance. Lift and thrust benefit from flexibility as well as from both passive and active wing deformation. At the same time, the active deformations are used for flight control. In this study, we investigate strategies of control moments generation in a bio-inspired flapping-wing micro air vehicle (FWMAV). In particular, we propose a method for active control and attitude stabilization by introducing a wing deformation through adjustable wing sweep. The control method is demonstrated on a tailless FWMAV with independent wing sweep modulation on each of its four wings. The actuation mechanism consists of an arm joint at the leading edge, about which the wings are swept. Forces from the servo actuation are transferred to the leading edge of the robot through strings. The actuated strings alter the wing sweep, which affects the roll and pitch movement via different combinations of string pulls. The effectiveness of the designed mechanism is being evaluated on the basis of tethered force balance tests and free flight tests. An advantage of the proposed mechanism is its lightweight design, which is crucial for small FWMAVs with stringent weight restrictions.

1 INTRODUCTION

Diversity in environmental conditions have forced airborne animals to perfect their flight manoeuvres. Many natural flyers rely on wing morphing to achieve various flight maneuvers such as making swift turns or dodging an obstacle. During flapping, birds tend to fold their wings at the beginning of upstroke to reduce the counter productive forces by not only decreasing their wing area [1], but also reducing the inertial forces [2]. Furthermore, a re-configurable wing geometry allows changing the lift and drag coefficients. Gliding birds in particular take advantage of their swept wings at high gliding speeds to minimize drag [3], but during take-off and landing, their extended wings maximize drag and lift accordingly [4]. Morphed wings can also produce lower aerodynamic load, decreasing the risk of flow separation [5, 6].

The idea of wing morphing has encouraged many UAV (Unmanned Aerial Vehicle) researchers to design aircraft and bio-inspired prototypes of aerial robots that enable wing morphing by either altering the profile or planform of the wing [7, 8, 9]. Inspired by bats and birds, Stowers [10] has adapted passive wing morphing by introducing a mechanism with wrist joints to fold and unfold the wing. This research has also led to a great contribution to the understanding of the physics of wing morphing. RoboSwift, a MAV that mimicks the agile swift bird, uses active wing sweep to ensure highly efficient gliding flight at various flight speeds [11]. The prototype can quickly change its wing area, sweep, slenderness and camber by folding its ”feathers” backward while its flight control and stability is maintained with use of a tail.

The development of tail-less flapping wing MAVs that use morphed wings for control is even more challenging. For instance, BatBot is equipped with the multiple-degree-of-freedom wings which are covered using a flexible membrane [12] in order to enhance the number of possible maneuvers. However, the complex actuation mechanism represents a significant weight penalty. Due to limited lift production, the vehicle’s flight maneuvering capabilities could only be demonstrated in fast forward, descending flight. The Nano Hummingbird by Keennon [13] uses the concept of wing twist modulation and rotation to achieve control and stability of the robot. The prototype also demonstrates an outstanding flight performance and maneuverability, allowing hovering as well as fast forward flight. However, the mechanism used to control the Hummingbird is highly complex. Several more recent designs lead to successful flight, but with less complex mechanisms [14, 15]. In terms of complexity, arguably the simplest design, dubbed ”quad-thopter”, was introduced in [16]. The quad-thopter is named after the quadrotor, where the rotors are replaced by (in this case double) wings. The design and control of this vehicle is rather straightforward. However, there is a weight and miniaturization penalty involved in having 4 gearboxes, motors, etc. Despite the successes in the literature, the search for straightforward, lightweight and high-performance control designs continues, also with further miniaturization in mind.

Regardless of the different designs, as of yet, there is no existing flapping wing MAV that utilizes swept wing technique to achieve autonomous flight. In this article, we present a tailless FWMAV that uses a novel method to attain flight

*D.A.Olejnik@tudelft.nl
†aadithyasujit@gmail.com

control by means of a wing sweep deformation technique. The system benefits from simplicity arguably, the motor just provides propulsion, and a servo the attitude control.The prototype is equipped with an autopilot for active stabilization and an autonomous flight. The FWMAV design is based on the Delfly flapping-wing vehicle developed at TU Delft [17]. We have kept its reliable flapping mechanism, but adjusted the leading edge design such that they can be bent by servo-actuated strings. The employed solution eliminates the need for a tail or any other additional control surfaces such as an elevator or rudder, thereby ensuring a lightweight design and reduction in the size of the FWMAV. By means of wing deformation, control moments can be actively generated and the developed MAV can exhibit close to hover flight. Preliminary tests show the capability of stable flight for 12 seconds.

2 Design and Materials

The starting point of this research is the Delfly bio-inspired MAV that can fly forward, backward for a short duration and even hover. Two actuated control surfaces on the conventional tail, the elevator and rudder generate the pitching and a combined yawing and rolling moments respectively. Due to the tailed design, this platform benefits from passive stability. A characteristic feature of the vehicle is the clap-and-peel mechanism of wings which induces additional thrust. For a more detailed information of the Delfly Project, the reader is referred to de Croon et al. (2016) [18].

Since the Delfly has proven to be a reliable platform allowing repeatable experiments, further development of the platform is mainly focused on the improvements that can be easily integrated with the existing MAV and allow to improve its controllability or limit power requirements. Here we present a tailless FWMAV (Figure 1) with bio-inspired wing sweep actuation mechanism.

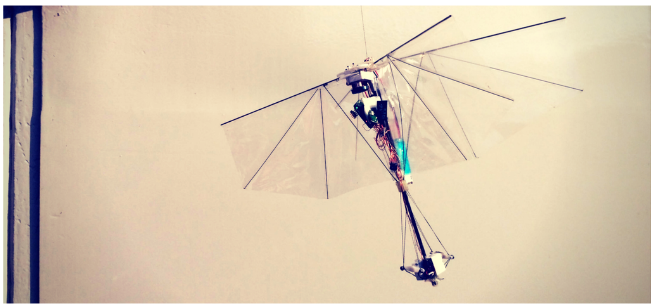


Figure 1: The proposed tailless flapping wing MAV with bio-inspired wing sweep actuation mechanism.

The prototype is equipped with a Lisa MXS autopilot [1], which consist of a 168 MHz STM32F4 microprocessor with 1 MB of flash memory. The board provides features such as a pressure sensor and three-axis gyros, accelerometers and magnetometers. To restrict the influence of the high frequency vibration and noise, polyurethane foam and depron were placed between the autopilot and the fuselage. The active stabilization and control are handled through the open-source autopilot software, Paparazzi UAV [2]. The pilot's commands are collected via radio link by DelTang Rx31 micro receiver. The Mi-3A brushless electronic speed controller (ESC), flashed with BLHeli firmware, is used to drive the BLDC motor. A pair of rotational servos (HobbyKing HK-5330 Ultra-Micro Digital Servo) capable of producing a torque up to 0.17kg is used as actuators, these actuators are coupled to their respective wing/wing pairs by means of low stretch string. This string has a line thickness of 0.12 mm and is made from Dyneema fibres making it ideal for high power pulling application without stretching. A high density-low weight Turnigy nano-tech LiPo battery with a 160 mAh capacity is used as the energy source for the system of electronics.

2.1 Design Concepts

In order to achieve greater agility of the prototype, we have come up with two viable solutions that utilize the wing sweeping technique to control the attitude of the MAV. In the first solution, by taking inspiration from nature, we introduce a wrist joint as a part of the leading edge of the wing which mimics a bat wing morphology (Figure 2). The hinge part was printed using a Multi-Jet Modeling process from a UV-cured acrylic polymer.

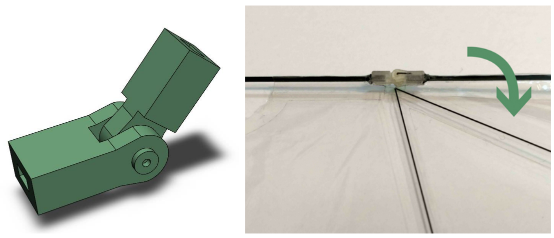


Figure 2: CAD model of the hinge (left), the hinge placed on the leading edge of the wing (right).

The hinge is positioned near the crossing of wing stiffeners at the leading edge. The leading edge carbon rods bend such as to increase wing sweep. In its second version, changes have been made to the structure to enhance the elasticity close to the hinge by adding thin steel rods. This slightly increased the capability of the wings. Downside of the solution is the fragility of resin based materials, which in free flight would be a risk. Due to these problems, decision was made to test another prototype.

An alternate solution to increase the magnitude of deviation along the wing is to vary the stiffness of the leading edge by either changing the cross sectional profile or thickness at a specific location. This section is characterized by decreased stiffness about which the leading edge bends. To achieve this,

[1] https://wiki.paparazziuav.org/wiki/Lisa/MXS_v1.0

[2] http://wiki.paparazziuav.org/

a small slit is milled on the leading edge rod using a CNC machine. The rod is then manually delaminated near the slit and the extent of delamination is restricted using shrink tubes (Figure 3). On testing different lengths of introduced gap, it was found that with increase in length, the required force to actuate the wing is minimized. The downside of an increased length is a faster deterioration of the material. The location of the delaminated section along the leading edge is chosen to be as close as possible to the swing arm (5 mm form the root) so that almost the entire wing can be bent. To externally actuate the wing bending, an actuation system inspired by our previous work on wing tension modulation [19] is used. As shown in figure 4, the system consists of rotational servos and strings that are attached to the leading edges. When the servo arm is actuated, it tensions the string which in turn forces the wing to bend in the desired direction. The advantage of this system being that the servo only needs to pull in one direction. When the servo arm is released, the leading edge naturally unbends itself.

Figure 3: Process of delamination of the carbon fiber rods.

The idea behind the mechanism is that whenever a wing is bent by pulling the strings, that particular wing alters its profile due to slackening. This effect causes the wing to produce less thrust relative to the other wings causing an imbalanced force production thus resulting in a control moment. By means of this mechanism, differential control can be performed by bending the leading edges of the wings for creating moments to control and navigate Delfly. Another effect is that, while bending a wing about its fuselage, the thrust vector is tilted such that it remains perpendicular to the leading edge. This vectoring of thrust on the bent wing can alter the moments generated and direction of the net thrust produced.

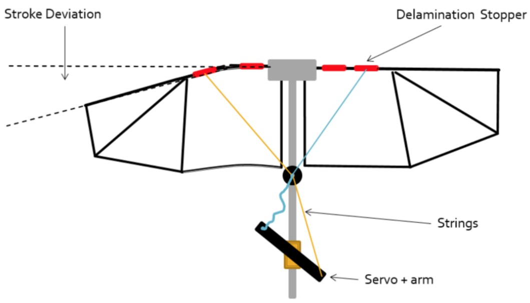


Figure 4: Bending of the leading edge using servo and string based actuation mechanism from the front view.

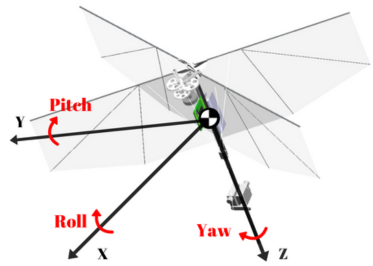


Figure 5: The axis convention used to describe the robot rotations.

To describe the body rotations, we adopt an aerospace coordinate system according to Figure 5. The proposed actuation scheme with differential control of wing bending is shown in Table 1. For instance, when both wings on the left side are actuated, the thrust produced by the actuated pair is significantly lower relative to that of the right side. This difference in thrust produced between the two sides generates a moment about the center of gravity (cg) which steers the flapper to the left side resulting in a left roll. Likewise, pulling the right wing pair will result in a right roll. Similarly, when the bottom wings of either side of the flapper are actuated, the corresponding wing pair loses thrust relative to the top pair, resulting in a moment that pitches the flapper downwards. Likewise, pitching up moment will be generated when the top pair of the wings are actuated. Although our hypothesis does not hold strong for the yaw command, the flapper is expected to generate yaw when the adjacent pair of wing are actuated. The possible reason for this could be that when the wings are actuated, the slackness of the wing causes the foil of the wing to freely flex during one stroke while in the return stroke, the foils deformation is restricted by the string. This alternately occurring phenomena is expected to result in a yaw moment. The clap-and-peel mechanism of wings is less effective for the considered maneuver. The proposed arguments have been proved in by the experimental study shown in Chapter 4.

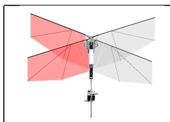
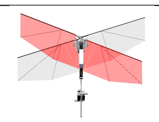
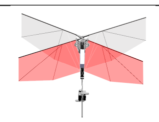
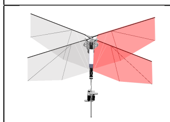
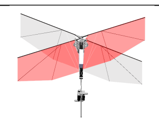
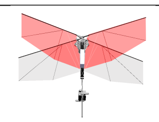
 Roll Left	 Yaw Left	 Pitch Down
 Roll Right	 Yaw Right	 Pitch Up

Table 1: Control moments generation scheme with respect to the used combinations of a wing pulling. The flapper is shown from rear view and actuated wings are colored in red.

3 Experimental Setup

To determine and validate the capabilities of the prototype, we performed two tests: tethered and free flight. Further explanation of the experimental setup is presented in the following chapter with main focus on force balance tests.

3.1 *Tethered force balance tests*

In order to validate the generated moments, the prototype was mounted onto a 6 axis force transducer - ATI Nano-17 Titanium. The measurement setup logs the forces and moments along the 3 axis. Additionally, flapping frequency and power consumption is recorded via measurements of current, voltage, and counting of the brushless motor polarity changes. The flapper was clamped on the force sensor close to the cg in a yz plane.

A servo tester connected to the ESC was used to adjust the flapping frequency. All the experiments were conducted for a range of frequencies varying from 6 z to 20 Hz; the vehicle hovers at about 15 Hz. This assumption is made based on the data of a similar sized MAV used in the past [18]. The data acquisition was handled by the NI cRIO-9024 controller with a FPGA.

Data processing was performed using MATLAB R2017b software. The obtained raw data of forces and moments was filtered using Chebyshev Type II low-pass filter with -80dB attenuation of the stopband. To prevent time shift of data, the forward-backward filtering technique was used via filtfilt function of MATLAB [14]. The 50 Hz cut-off frequency was selected based on assumption that we should keep at least first two harmonics of Z-force power spectral density (Figure 6) due to the suggestion that the two first peaks are related to the aerodynamic forces production [20].

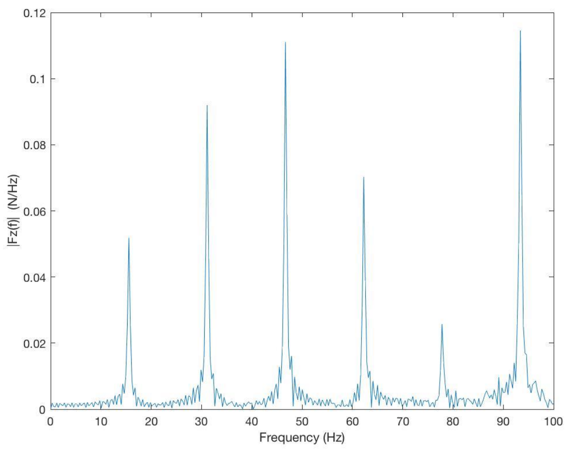


Figure 6: Single-Sided Amplitude Spectrum.

3.2 *Free flight test*

To achieve stable flight, the attitude of the 19.76-gram prototype was autonomously controlled using a PD controller with attitude feedback from the on-board IMU of the Lisa MXS autopilot. The P and D gains were hand-tuned using a trial and error method, until acceptable performance was achieved. As a starting step, the gains were initialized to the values of similarly sized vehicles and tuned until a stable flight is achieved.

For the power source, a single cell (1S) high density and low weight LiPo battery is used. Two batteries with similar specifications but with varying discharge rates were used for the Delfly in free flight. The position of the battery (being the most significant weight on the prototype) is also very crucial for determining the cg location of the Delfly, which in turn determines the static and dynamic stability [21].

The total moment generated by the flapper depends on the relative distance of the cg and the application point of the acting forces. When the battery was placed at the bottom of the fuselage, the dynamic stability of the flapper improved but the control effectiveness decreased. When a roll or pitch command is given, the normal or side drag force opposing the body motion acts on a large moment arm and results in a significant opposite (counter) moment to that of the desired control moment [22], which decreases the overall effectiveness of the control.

When the battery was placed at the top of the flapper, high control effectiveness was achieved but leaving the system dynamically unstable. Because the moment arm of the body-motion induced drag forces is now in the opposite direction, this drag-based moment is in the same direction as the desired control moment, virtually increasing the control effectiveness, but (statically) destabilizing the system.

In the first flight trials, the vertical location of the battery was being adjusted to find a suitable vertical cg location. We ended up with a battery placed roughly near the quarter chord point, which was a good compromise between effective control and stability, and is in agreement with recommendations based on theoretical models given in [21].

4 Results & Discussion

4.1 *Clamped force data*

First, the tethered force balance tests have been conducted. The moments, recorded in the force balance reference frame, was transformed to the vehicle cg.

As shown in Figure 7, the flapper produces a mean force Fz of -0.29 N when no actuation of the wing sweeping mechanism is applied. The generated force is sufficient enough to support the flight of the 19.76 gram FWMAV. It can be observed that the mean force Fx is very close to 0 N. This is due to the symmetric flapping motion of the front and rear wings that flap in counter sense, resulting in a cancellation of the Fx force produced by each individual wing. In the hover case (zero free stream), the flapper purely relies on the Fz force to stay in air.

Although we thought that a yaw moment could be generated, the experiment showed no significant yaw moment

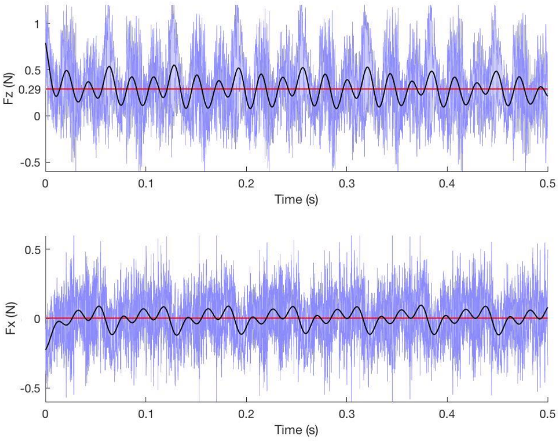


Figure 7: The tethered force balance measurements of Fz and Fx forces. The blue and black lines represents raw and filtered data respectively, while the red line indicates the mean value of the filtered data.

production. When wings were actuated (see Table 1 - yaw maneuver) the flapping frequency of DelFly dropped significantly, losing thrust below the equivalent weight of the DelFly. This was due to increased friction at the cross over region of the swing arms that led to loss of flapping frequency and thrust.

However, the roll (Mx) and pitch (My) moments measurements are generally in good agreement with the assumed scheme of control moments generation. Comparison of rolling and pitching moments characteristics during the flapping motion for different actuations (Table 1 of the swept wings can be seen in Figure 8). For the sake of clear comparison no actuation mode is also displayed.

When the shape of the Fx curve in Figure 7 is closely observed, it can be seen that each flapping cycle consist of two minor peaks and one major peak. When the position of the leading edge was studied using a hall sensor with comparison to the respective forces and moments, it was found that the major peak occurs during the peel action or the outward stroke and the following minor peaks occurs during the the clapping action and possibly at the stroke reversal when the wing flexes pushing more air. For more details, the reader is referred to [14].

This trend is very similar in the Mx moment plot showed in Figure 8, the plot clearly depicts these peaks for no actuation and left actuation and to some extent in the right actuation curve. The possible reason for the absence of third minor peak in the right actuation curve could be due to the interaction of wing and string when actuated does restricts the natural flexing of wing at stroke reversal.

It can be noticed, that the average effective moment generated about the roll axis is higher than pitch axis. Although

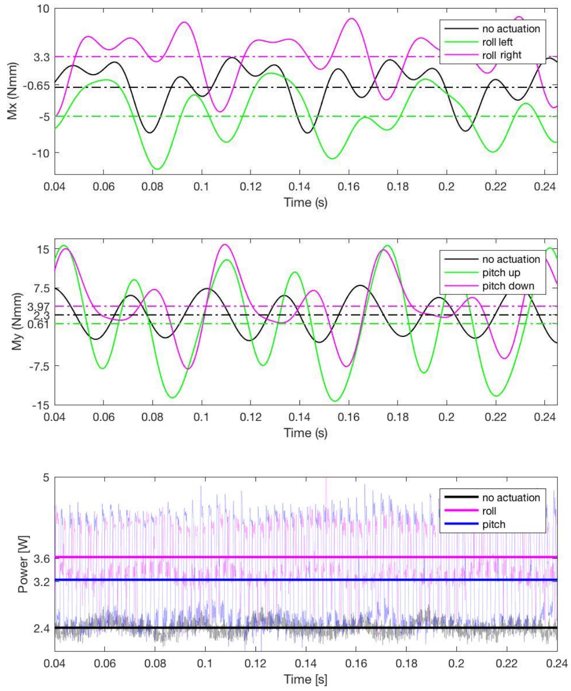


Figure 8: Comparison of rolling and pitching moments characteristics (top and center) and mean average power (bottom) during the flapping motion for different actuation scheme.

the displacements due to actuation for all the 4 independent wings were calibrated to be of same magnitude, and thus have a similar effect on the modulation of each wing's thrust force, the moment arm of the thrust force is larger in the roll case ($L/2cos(\phi/2)$) than in the pitch case ($L/2sin(\phi/2)$), which explains the higher roll moments. Here we assumed that the wing thrust force lies near the mid-span of the wing with a length L, ϕ is the flapping angle measured with respect to the y axis.

4.2 *Power Requirements*

On testing the power requirements for an actuated and non-actuated system, it was observed that the flapper uses a lot more power when actuated. Although the servo limits were set as to not cause any saturation, the system draws additional current to bend the wing and hold them in the desired position. Additionally, bending of the wing increases the friction between the hinges resulting in increased motor load. This decreases the flapping frequency. In Figure 8, it can be observed that for a given time of 0.25 secs, the actuated plot comprises of three flapping cycles while the non-actuated plot comprises of four flapping cycles. This implies that the actuated wing flaps at 12-13 Hz while a non-actuated wing flaps around 15 Hz. From Figure 8, actuation of flapper increases the power requirement from 2.4 W to 3.6 W. It can also be observed that the roll actuators draw more power than the pitch due to the design of the swing arm and fuselage interface.

4.3 *Flight Testing*

For a further confirmation of our findings, we have also performed free flight experiments. During the first test, with a slightly used Hyperion battery (180 mAh) as a power source, for a given input throttle command, the altitude of the Delfly was slowly decreasing with auto stabilization. However without auto stabilization, the Delfly seemed to climb for the same throttle command. This indicates that with the auto stabilization, the system draws more current than the power source can provide. After replacing the power source with the a brand new and 1 gram lighter battery (Turnigy- 160 mAh), the problem of decreasing thrust was addressed. Due to lower mass the current draw has decreased. However, this would reduce the overall flight time due to the lower capacity of the chosen battery. It was also observed that the structural support for the autopilot played a crucial role in achieving a hovering free flight without saturating the on board sensors of the autopilot. Adding a depron and PU foam between the fuselage and the autopilot absorbed most of the noise caused by the vibration of flapper. A standard state estimation and loop within the paparazzi software was used and parameters related to roll and pitch were adjusted. The gains of the controller were tuned to stabilize the attitude of the flapper.

With sufficient current available and a tuned controller, the prototype could self stabilize in hover condition for 12 seconds (the three first frames on Figure 9) while gradually building oscillations and decreasing its altitude indicating loss of thrust (the fourth and fifth frames on Figure 9) due to low battery level. These oscillations of the flapper along the pitch axis indicates insufficient power to the servo actuators. The actuators draw more current either while trying to hold an actuated position or at the extreme actuated position. This results in shortage of power to the motor which eventually decreases the flapping frequency, and results in a loss of thrust (the sixth frame on Figure 9). (The supplementary video is available online at `https://www.youtube.com/playlist?list=PLwJoNhf07bFJefdur7OhHrAzIU3hCMZOq`)

5 Conclusion

Inspired by birds and bat flight we developed a flapping wing MAV that uses wing sweep modulation for active control and stabilization. We carried out the tethered force balance tests and free flight experiments to validate the platform performance as well as to confirm the assumed scheme of control moments generation resulting from actuation of various wing combinations. An advantage of the proposed mechanism is its lightweight design and simplicity.

The objective of future work is to extend this analysis of alternative bending points and possibly design new actuation mechanism that will also allow to generate a yaw moment. The design of the hinge and the sandwiched swing arm configuration can be improved to not only handle larger magnitudes of actuation but also to reduce the frictional loss. Thereby, minimizing the power needed during actuation. In addition, the duration of the final flight time can be increased by looking more into the power management of the system. A possible solution could be to use an independent power source for the actuators to ensure that thrust production is unaffected during extreme actuation.

Figure 9: Time frames of a free flight test.

Acknowledgements

This work is part of the Open Technology Programme with project number 15039, which is financed by the Netherlands Organisation for Scientific Research (NWO).

Netherlands Organisation
for Scientific Research

References

[1] Florian T Muijres, Melissa S Bowlin, L Christoffer Johansson, and Anders Hedenström. Vortex wake, downwash distribution, aerodynamic performance and wingbeat kinematics in slow-flying pied flycatchers. *Journal of The Royal Society Interface*, 9(67):292–303, 2012.

[2] Daniel K Riskin, Attila Bergou, Kenneth S Breuer, and Sharon M Swartz. Upstroke wing flexion and the inertial cost of bat flight. *Proceedings of the Royal Society of London B: Biological Sciences*, 279(1740):2945–2950, 2012.

[3] D Lentink, UK Müller, EJ Stamhuis, R De Kat, W Van Gestel, LLM Veldhuis, Per Henningsson, Anders Hedenström, John J Videler, and Johan L Van Leeuwen. How swifts control their glide performance with morphing wings. *Nature*, 446(7139):1082, 2007.

[4] Brett Klaassen van Oorschot, Emily A Mistick, and Bret W Tobalske. Aerodynamic consequences of wing morphing during emulated take-off and gliding in birds. *Journal of Experimental Biology*, pages jeb–136721, 2016.

[5] Anna Carruthers, Graham Taylor, Simon Walker, and Adrian Thomas. Use and function of a leading edge flap on the wings of eagles. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, page 43, 2007.

[6] Rudolf Dvořák. Aerodynamics of bird flight. In *EPJ Web of Conferences*, volume 114, page 01001. EDP Sciences, 2016.

[7] Daniel Grant, Mujahid Abdulrahim, and Rick Lind. Flight dynamics of a morphing aircraft utilizing independent multiple-joint wing sweep. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, page 6505, 2006.

[8] John Flanagan, Rolf Strutzenberg, Robert Myers, and Jeffrey Rodrian. Development and flight testing of a morphing aircraft, the nextgen mfx-1. In *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1707, 2007.

[9] AA Wissa, Y Tummala, JE Hubbard Jr, and MI Frecker. Passively morphing ornithopter wings constructed using a novel compliant spine: design and testing. *Smart Materials and Structures*, 21(9):094028, 2012.

[10] Amanda K Stowers and David Lentink. Folding in and out: passive morphing in flapping wings. *Bioinspiration & biomimetics*, 10(2):025001, 2015.

[11] The RoboSwift Team. *RoboSwift*, (accessed 28 July 2018).

[12] Alireza Ramezani, Xichen Shi, Soon-Jo Chung, and Seth Hutchinson. Bat bot (b2), a biologically inspired flying machine. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 3219–3226. IEEE, 2016.

[13] Matthew Keennon, Karl Klingebiel, and Henry Won. Development of the nano hummingbird: A tailless flapping wing micro air vehicle. In *50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 588, 2012.

[14] M Percin. *Aerodynamic mechanisms of flapping flight*. PhD thesis, TU Delft, Delft University of Technology, 2015.

[15] Frederik Leys, Dominiek Reynaerts, and Dirk Vandepitte. Outperforming hummingbirds load-lifting capability with a lightweight hummingbird-like flapping-wing mechanism. *Biology open*, 5(8):1052–1060, 2016.

[16] Christophe De Wagter, Matej Karásek, and Guido de Croon. Quad-thopter: Tailless flapping wing robot with 4 pairs of wings. *9th international micro air vehicles*, 2017.

[17] Guido CHE de Croon, MA 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, 2012.

[18] G. C. H. E. de Croon, Mustafa Percin, B. D. W. Remes, Rick Ruijsink, and C. De Wagter. *The DelFly - Design, Aerodynamics, and Artificial Intelligence of a Flapping Wing Robot*. Springer Netherlands, 2016.

[19] R.M.J. Janssen. Attitude control- and stabilisation moment generation of the delfly using wing tension modulation, 2016.

[20] JV Caetano, M Percin, BW van Oudheusden, B Remes, C De Wagter, GCHE de Croon, and CC de Visser. Error analysis and assessment of unsteady forces acting on a flapping wing micro air vehicle: free flight versus wind-tunnel experimental methods. *Bioinspiration & biomimetics*, 10(5):056004, 2015.

[21] Matěj Karásek. *Robotic hummingbird: Design of a control mechanism for a hovering flapping wing micro air vehicle*. PhD thesis, Universite Libre de Bruxelles, 2014.

[22] Bo Cheng and Xinyan Deng. Translational and Rotational Damping of Flapping Flight and Its Dynamics and Stability at Hovering. *IEEE Transactions on Robotics*, 27(5):849–864, oct 2011.

Forward Flight Control Analysis of Bioinspired Flapping-Wing Micro Air Vehicle

Jong-Wan Lee, Jong-Seob Han, and Jae-Hung Han*
Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea

ABSTRACT

The forward flight control of FWMAV is studied at varying reference flight speeds. The semi-empirical quasi-steady aerodynamic model including effects of advance ratio is used to calculate the aerodynamic load on wings and control gain matrices are obtained using LQR method.

1 INTRODUCTION

To utilize the agility of flapping-wing micro air vehicle (FWMAV), many studies on the dynamic characteristic of flapping-wing MAV have been conducted. However, it is difficult to organize the control system for FWMAV under various flight conditions because of the high non-linearity and instability of FWMAV itself. Moreover, the linearization method, which has been widely used in the previous control studies of FWMAV, is valid only around the equilibrium points. It means that the controller obtained by using the linearized system equation around a specific equilibrium point may not properly work at other flight conditions. The quasi-steady aerodynamic model, which is easily applicable to dynamic analysis of FWMAV, needs to be revised in case of forward flights because the forward flight speed affects the stability of Leading-Edge Vortex (LEV) on the wing. There could be some error between the real aerodynamic loads and the results of the conventional quasi-steady aerodynamic model when FWMAV moves in forward direction. This study uses a semi-empirical quasi-steady model which is more accurate in forward flight than conventional quasi-steady model.

This study investigates the forward flight control of FWMAV at varying reference flight speeds.

2 MATERIALS AND METHOD

2.1 System modelling and coordinate definition

The FWMAV model consists of the body and one pair of wings. The morphologies of its body and wings are based on the Hawkmoth *Manduca sexta* by referring to the studies of O'Hara et al [1] and Ellington [2]. The wings are assumed to be massless and rigid. In addition, aerodynamic effects on the body are ignored.

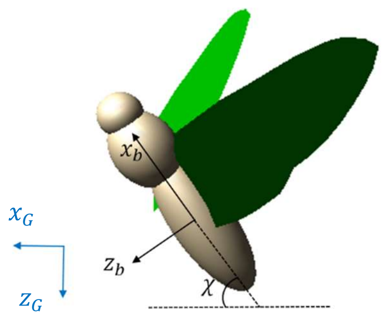


Figure 1 - FWMAV(Manduca sexta)model.

Figure 1 shows FWMAV model used in this study. The body-fixed coordinate whose origin located at the center of mass of body is expressed by using subscript b. Global coordinate is expressed by subscript G. Body pitch angle written as χ means the angle between the body-fixed coordinate and the global coordinate.

2.2 Definition of wing kinematics

The wing kinematics are described using three angles, stroke position angle (ϕ), feathering angle(α)and deviation angle(θ). Each rotational angle is set as follows:

*contact email: jaehunghan@kaist.edu

$$\phi(t) = \phi_0 - \phi_{amp} \sin(2\pi f t)$$
$$\alpha(t) = \alpha_0 - \frac{\alpha_{amp}}{\tanh(C_\alpha)} \tanh(C_\alpha \cos(2\pi f t)) \qquad (1)$$
$$\theta(t) = 0$$

2.3 Aerodynamic model

To describe the aerodynamic load on the flapping wing, the semi-empirical quasi-steady aerodynamic model from the study of Han et al. [3] is used. This model compensates the influence of the advance ratio J by defining aerodynamic coefficients as functions of the angle of attack and advance ratio J both. These coefficients (Equation 2) were obtained by conducting 147 individual experiment cases at each advance ratio J value and angle of attack value. For more details, refer to Han et al.[3]

$$\begin{aligned} C_L(\alpha, J) &= K_{p,L}(J)\sin(\alpha)\cos^2(\alpha) \\ &\quad + K_{V,L}(J)\sin^2(\alpha)\cos(\alpha) \\ C_D(\alpha, J) &= K_{P,D}(J)\sin^2(\alpha)\cos(\alpha) \\ &\quad + K_{V,D}(J)\sin^3(\alpha) \\ C_M(\alpha, J) &= K_{P,M}(J)\sin^2(\alpha)\cos(\alpha) \\ &\quad + K_{V,M}(J)\sin^2(\alpha) \end{aligned} \qquad (2)$$

$$J = \frac{U_\infty}{U_{tip}} = \frac{u_G}{4\phi_{amp} R f} \qquad (3)$$

The advance ratio is defined as the ratio of forward flight speed to mean wing tip velocity as shown in Equation 3.

2.4 Equation of motion and linearization

Equation 4 is the 3-DOF longitudinal equations of motion expressed in body-fixed coordinate. Here, the X, Z and M are periodic aerodynamic forces and moment induced by wing motion. u, w and q are the velocities of x and z direction of the body-fixed coordinate and the body pitch rate, respectively.

$$\begin{aligned} X - mg\sin\chi &= m(\dot{u} + qw) \\ Z + mg\cos\chi &= m(\dot{w} - qu) \\ M &= I_{yy}\dot{q} \end{aligned} \qquad (4)$$

Equation 4 can be linearized around equilibrium points using the cycle-average method and the small perturbation theory. Equation 5 is the linearized equation of motion expressed in matrix form.

$$\begin{bmatrix} \Delta\dot{\bar{u}} \\ \Delta\dot{\bar{w}} \\ \Delta\dot{\bar{q}} \\ \Delta\dot{q} \end{bmatrix} = \begin{bmatrix} \frac{\bar{X}_{\bar{u}}}{m} & \frac{\bar{X}_{\bar{w}}}{m} & \frac{\bar{X}_{\bar{q}}}{m} - \bar{U}_\infty \sin\chi & -g\cos\chi \\ \frac{\bar{Z}_{\bar{u}}}{m} & \frac{\bar{Z}_{\bar{w}}}{m} & \frac{\bar{Z}_{\bar{q}}}{m} + \bar{U}_\infty \cos\chi & -g\sin\chi \\ \frac{\bar{M}_{\bar{u}}}{I_{yy}} & \frac{\bar{M}_{\bar{w}}}{I_{yy}} & \frac{\bar{M}_{\bar{q}}}{I_{yy}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta\bar{u} \\ \Delta\bar{w} \\ \Delta q \\ \Delta\Theta \end{bmatrix} \qquad (5)$$

More details on process and the assumption of the linearization can be found in the study of Kim et al. [4].

2.5 Trim search

Since the linearization of nonlinear system is conducted near equilibrium points, equilibrium points should be found first. To find out the trim conditions, which make the FWMAV model slightly oscillate around an equilibrium point, we employed the gradient-based trim search algorithm from the study of Kim et al. [4]. This trim search algorithm runs the dynamic simulation of FWMAV during the one flapping period iteratively as changing the initial conditions until the mean velocity and excessive force and moment conditions are satisfied. The excessive forces and moment are compensated by using inverse matrix of control effective matrix B that consists of the partial derivative of force and moment to each control parameters. The control effective matrix B contains gradient information of the aerodynamic force and moment with respect to each control parameters. In this study, wing stroke frequency (f), mean stroke angle (ϕ_0) and mean feathering angle(α_0) were chosen as control parameters. The more precise and detailed expression and procedure can be referred to Kim et al. [4]. For dynamic simulation, the ode45 solver with 10^{-4} time step size in MATLAB® Simulink was used.

$$B = \begin{bmatrix} \frac{\partial \overline{X_G}}{\partial f} & \frac{\partial \overline{X_G}}{\partial \phi_0} & \frac{\partial \overline{X_G}}{\partial \alpha_0} \\ \frac{\partial \overline{Z_G}}{\partial f} & \frac{\partial \overline{Z_G}}{\partial \phi_0} & \frac{\partial \overline{Z_G}}{\partial \alpha_0} \\ \frac{\partial \overline{M}}{\partial f} & \frac{\partial \overline{M}}{\partial \phi_0} & \frac{\partial \overline{M}}{\partial \alpha_0} \end{bmatrix} \quad (6)$$

To find the trim condition at each forward speed, the reference body pitch angle should be determined as a function of forward speed. Since it was well known that body of FWMAV tilts more toward forward flight direction as it moves faster, which means χ decreases as speed increases from the study of [5], we determined the body pitch angle at each forward flight speed as simple linear function as described in Equation 7. The body pitch angle is 70 degree at hovering state and decreases linearly at rate of 20 $\deg/(m/s)$.

$$\chi(\deg) = 70 - 20u_G \quad (7)$$

2.6 Optimal control gain matrices decision with LQR method.

At each trim condition, the linearized equations of motion with control effectiveness matrix B are obtained.

$$\begin{bmatrix} \Delta\dot{\bar{u}} \\ \Delta\dot{\bar{w}} \\ \Delta\dot{\bar{q}} \\ \Delta q \end{bmatrix} = A \begin{bmatrix} \Delta\bar{u} \\ \Delta\bar{w} \\ \Delta\bar{q} \\ \Delta\bar{\chi} \end{bmatrix} + B_u \begin{bmatrix} \Delta f \\ \Delta\phi_0 \\ \Delta\alpha_0 \end{bmatrix} \quad (8)$$

The detail terms for the matrix A can be found in Equation 5. B_u is the body-fixed coordinate and the normalized version of B is given in Equation 6.

$$B_u = \begin{bmatrix} \frac{\partial \overline{X_b}}{\partial f}/m & \frac{\partial \overline{X_b}}{\partial \phi_0}/m & \frac{\partial \overline{X_b}}{\partial \alpha_0}/m \\ \frac{\partial \overline{Z_b}}{\partial f}/m & \frac{\partial \overline{Z_b}}{\partial \phi_0}/m & \frac{\partial \overline{Z_b}}{\partial \alpha_0}/m \\ \frac{\partial \overline{M_b}}{\partial f}/I_{yy} & \frac{\partial \overline{M_b}}{\partial \phi_0}/I_{yy} & \frac{\partial \overline{M_b}}{\partial \alpha_0}/I_{yy} \\ 0 & 0 & 0 \end{bmatrix} \quad (9)$$

Since the uncertainties in dynamic modelling can make an offset, we set an integral state (Equation 10) to obtain the augmented equation (Equations 11 and 12).

$$X_I = \int_{t_{start\ of\ cycle}}^{t_{end\ of\ cycle}} y\, d\tau \quad (10)$$

$$\begin{bmatrix} \dot{X} \\ \dot{X}_I \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} X \\ X_I \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u \quad (11)$$

$$\dot{\tilde{X}} = \tilde{A}\tilde{X} + \tilde{B}u \quad (12)$$

Here, C matrix is defined as

$$C = \begin{bmatrix} \cos\chi_{ref} & \sin\chi_{ref} & 0 & 0 \\ -\sin\chi_{ref} & \cos\chi_{ref} & 0 & -U_\infty \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

By using the augmented version of the equation (Equation 12) and the weight matrices(R and Q), we could obtain the optimal gain matrix $\tilde{K}$. R is the weight matrix for the control input parameters and Q is the weight matrix for the states.

$$R = \begin{bmatrix} 100 & & \\ & 100 & \\ & & 100 \end{bmatrix} \quad (14)$$

$$Q = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 10^{-4} & & & & \\ & & & 0.01 & & & \\ & & & & 10 & & \\ & & & & & 10 & \\ & & & & & & 20 \end{bmatrix} \quad (15)$$

$\widetilde{K}$ can be divided into two parts, one is for the original states and the other is for the integral states.

$$\widetilde{K} = [K \ K_I] \quad (16)$$

$$u = -KX - \sum K_{I,i} X_{I,i} \quad (17)$$

Equation 17 shows how the control input u is determined for the next cycle. Here, since the control input is updated at every half period and the control gain matrix changes with the reference velocity, the integral state parts of Equation 17 is accumulated whenever the control gain is updated, instead of being calculated by multiplying $K_{I,i}$ to an integrated value over whole simulation time.

3 RESULT AND DISCUSSION

3.1 Trim search results

Figure 2 shows the trimmed wing kinematics results at each flight speed. The flapping frequency decreases overall as flight speed increases. On the other hand, the mean stroke angle and the mean feathering angle increase as the flight speed decreases. Increased mean stroke angle means that mid stroke line goes back with respect to the body. Increased mean feathering angle represents increased angle of attack at downstroke and decreased angle of attack at upstroke. This combinations of wing kinematic control parameters at each flight speed generate proper forces and moment to stay in trim condition. The gain matrix $\widetilde{K}$ is also obtained at each trim condition. Figure 3 and 4 show the components of K and K_I in Equation 16, respectively. The gain matrix $\widetilde{K}$ is set as functions of forward flight speed from the trim condition results by linearly interpolating between two nearest point.

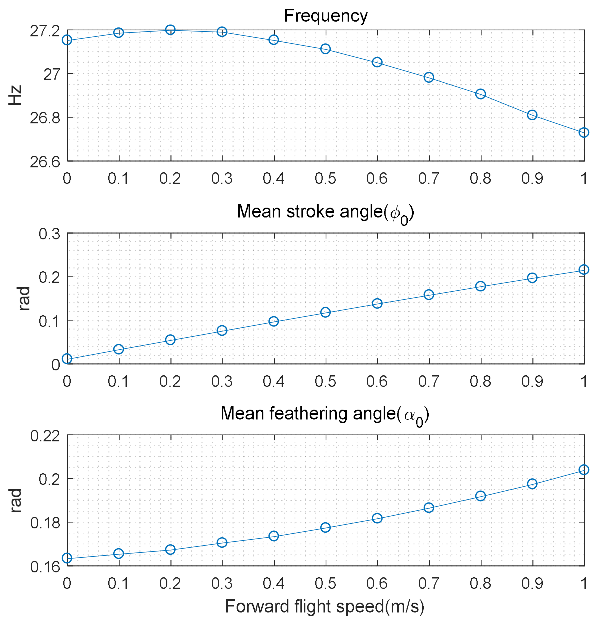


Figure 2 Trimmed wing kinematic parameters

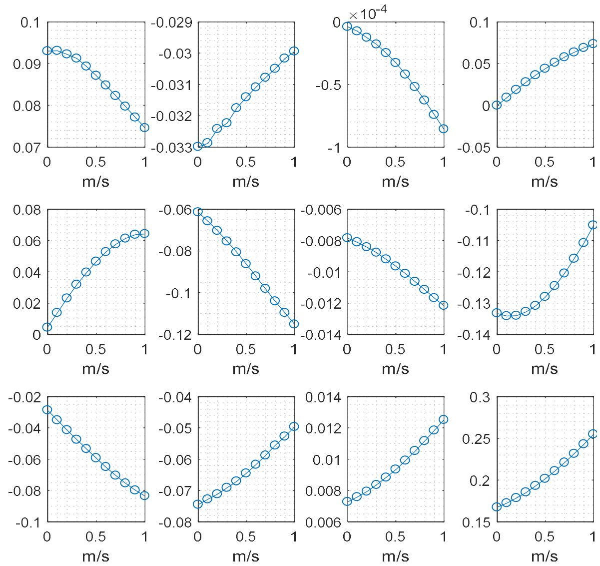


Figure 3 Component values of matrix K respect to flight speed

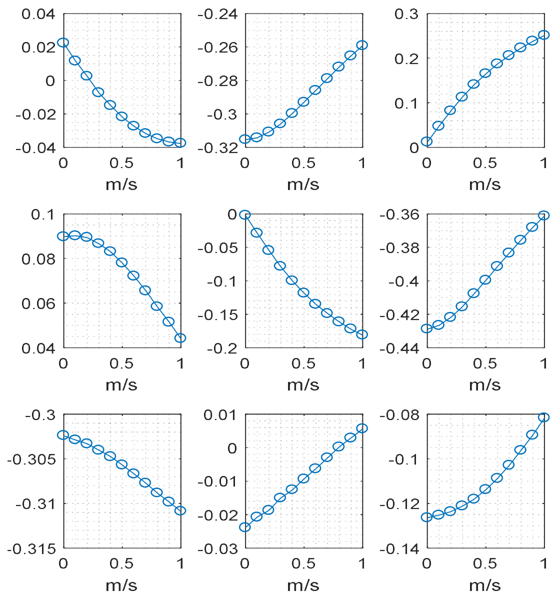


Figure 4 Component values of matrix K_I respect to flight speed

3.2 Flight control for varying reference velocity.

To see whether the designed gain-scheduled controller works well, the control simulation for the varying reference velocity was conducted. The reference velocity varies between 0m/s and 1m/s. Upper plot of Figure 6 Shows the velocity results. The red and blue lines represent the horizontal velocity reference and the horizontal velocity of FWMAV, respectively. The purple and yellow lines represent the vertical velocity reference and the vertical velocity of the FWMAV. The velocity results show that the model with controller follows the horizontal velocity reference well and the vertical velocity of model stays around 0m/s. The body pitch angle also follows the reference. There is a little deviation between the reference and body pitch angle of model during accelerating and decelerating. It seems that unwanted effects of accelerating and decelerating make the body pitch angle slightly deviated from the reference. Figure 5 shows the changes of the control parameters during the flight control simulation. The control parameters are changed in reasonably small range.

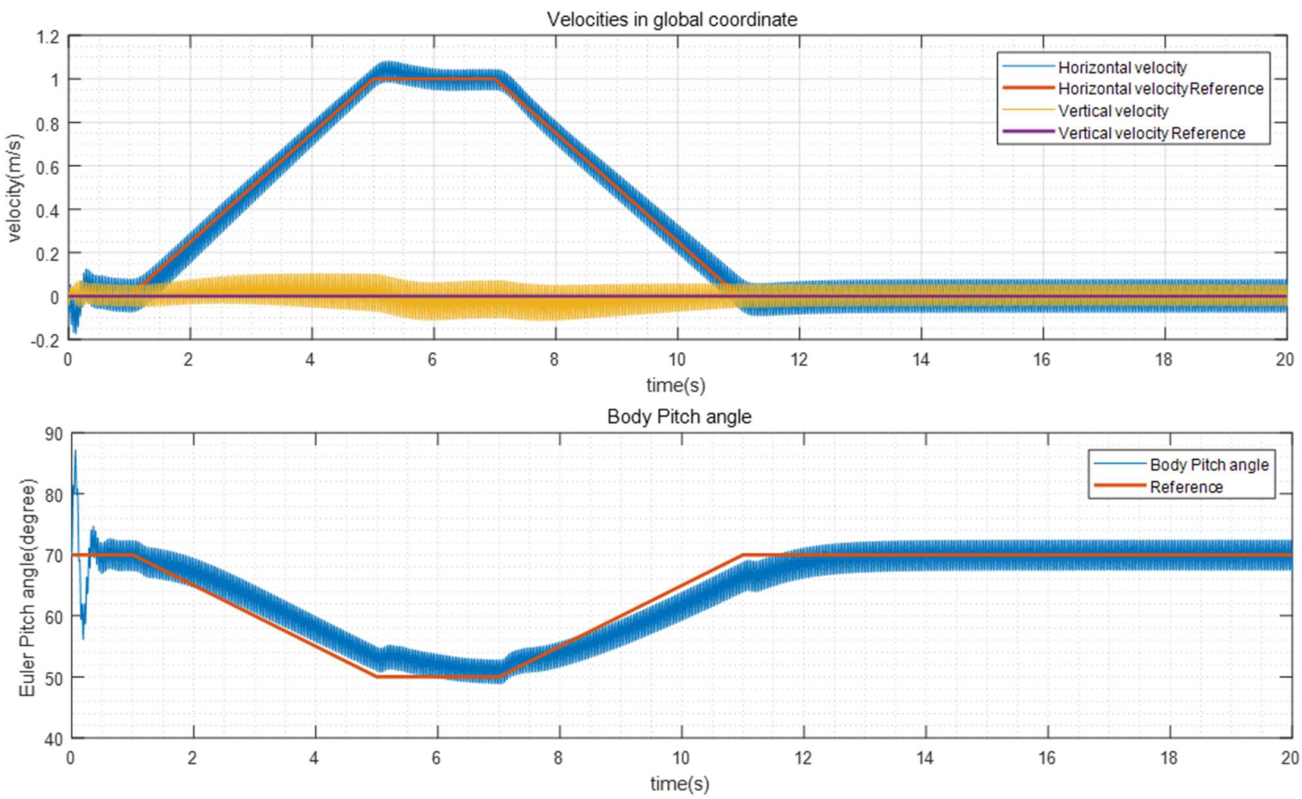


Figure 5 Control parameters during the flight control simulation

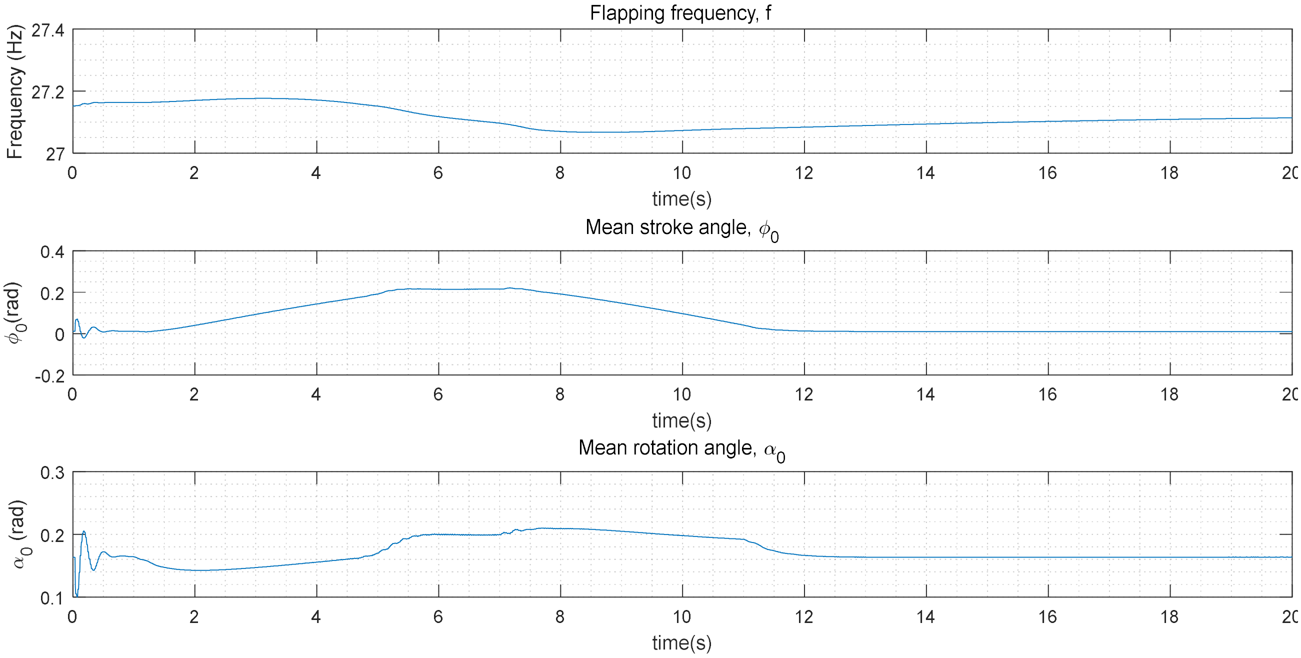


Figure 6 Flight control simulation results for varying reference velocity.

4 CONCLUSION

In this study, the forward flight control simulation of FWMAV is conducted. The equation of motion for a FWMAV is obtained using a semi empirical aerodynamic model and a rigid body dynamics approach. The control of FWMAV using wing kinematics variations is studied. It is shown that the designed gain scheduled controller works well for the speed tracking problems.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2017R1A2B4005676).

REFERENCES

[1] R.P. O'Hara, N. Deleon, and A. Palazotto. Structural identification and simulation of a MAV forewing. Composite Structures, 119, 315-321, 2015.

[2] C.P. Ellington. The Aerodynamics of Hovering Insect Flight. II. Morphological Parameters. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 305(1122): 17–40, 1984.

[3] J.-S. Han, J. Chang, J.-H Han, Jae-Hung. An aerodynamic model for insect flapping wings in forward flight. Bioinspiration & biomimetics, 12(3) : 036004 2017.

[4] J.K. Kim, J. S. Han, J. S. Lee, and J. H. Han Hovering and forward flight of the hawkmoth Manduca sexta: trim search and 6-DOF dynamic stability characterization. Bioinspiration & biomimetics, 10(5): 056012, 2015.

[5] C.P. Ellington. The Aerodynamics of Hovering Insect Flight. III. Kinematics. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 305(1122): 41–78, 1984.

Propulsion Performance Investigation of Bio-inspired Nano Rotor Base on Fluid-Structure Interaction

Z. Liu*, P. Che, B. Li, S. Zhao, T. Dang, C. Bu
School of Aerospace Engineering, State Key Laboratory for Strength and Vibration of Mechanical Structures, Xi'an Jiaotong University, No.28 Xianning West Road, Xi'an, Shaanxi 710049, China

ABSTRACT

The bio-inspired blade motion is introduced to improve the propulsive performance of nano rotor at an ultra-low Reynolds number. However, the complex flow interacts with the flexible composite blade structure resulting in the change of nano rotor propulsion performance and the vibration of blade structure. In this paper, a composite nano rotor with blade-pitch motion is investigated computationally with a computational solvers based on fluid-structure interaction. The finite element model for composite rotor is created and verified with a non-contact modal test. It is found that the simulation results matched well with the experimental results. Successively, the propulsive performance of a rigid nano rotor is studied. The propulsive performance of the nano rotor is analysed at different bio-inspired pitch frequency. Results show that the figure of merit of the bio-inspired pitch rotor increases because of the bio-inspired blade pitch motion. And it is also found that the improvement of the propulsive performance of the nano rotor varies with the pitch frequency. The propulsive performance of the flexible bio-inspired nano rotor is also studied with by using fluid-structure interaction method. It is found that the computational results for flexible nano rotor are lower than that for rigid nano rotor. It is evident that it is necessary to consider the flexibility of the composite nano rotor when investigating the propulsion performance of bio-inspired nano rotor. And the response of blade structure is also studied. Structural dynamic analysis shows that the blade structure vibrates with small amplitude. And two peak values are found at the rotation frequency and the fundamental frequency of the nano rotor structure.

1 INTRODUCTION

Rotary-wing Nano Air Vehicle (NAV) is a kind of small unmanned air vehicle powered with one or several rotors. NAV which has a maximum size of 7.5 cm and a minimum payload of 2 g is able to enter buildings, penetrate narrow entries, and transmit data without being detected at a low speed [1, 2, 3]. The nano rotor operating at a Reynolds number of lower than 20,000 is the main propulsion component of the rotary-wing NAV. At such a low Reynolds number, the aerodynamic performance of the nano rotor degrades and the figure of merit (FM) of the nano rotor drops as a result [4]. Liu et al. [5] measure the hovering performance of the propeller U-80 with diameter of 8 cm. Results show that the FM of the propeller is about 0.45 which is far lower than that of the full-scale helicopter. Therefore, how to improve the propulsion performance of the nano rotor is

*contact email: liuz@mail.xjtu.edu.cn

an important issue for the NAV design. However, the traditional method using the steady aerodynamic theory to optimize the propeller can only improve the propulsion performance limitedly.

Flying Insect whose size is comparable to NAV has a high aerodynamic efficiency. Lots of research shows that the unsteady mechanics induced by the flapping wing is the reason. The high-lift unsteady aerodynamic mechanisms include the clap-Fling, delayed stalled, rotational circulation and wake capture [6-8]. Some research is carried out by applying the high-lift unsteady mechanisms on the rotor to improve its aerodynamic efficiency [9, 10]. Fitchett [10] studies experimentally a conventional rotor, a rotor with powered blade flapping, and a freely rotating rotor with powered blade flapping. Results show that the maximum thrust increases by up to 15% and the torque required is reduced by up to 30% with conventional rotation plus powered blade flapping at up to 8 per rotor revolution (/Rev) at a reduced frequency of 0.6. Great enhancement can be found due to the blade flapping motion, but flapping motion requires more power to overcome the inertial force and aerodynamic load on the rotor blade due to the high rotational velocity. Koratkar [11] tests the blade pitch motion of a 22cm-diameter two bladed mciro rotor system featuring piezoelectricaly actuated controllable twist rotor blades to investigate the improvement in aerodynamic performance of micros rotors. The blade is motivated by two piezo-electrical beams allowing of changing the collective angle of blade. A 2.3° blade unsteady tip twist deformation is found resulting in an improvement of up to 11% at 24° rotor collective pitch. Results show that the blade pitch motion has an impact on the thrust once the stall onsets at the airfoil section of rotor blades. However, the crossing piezo-electrical beams limit the amplitude of oscillation. Ellington [12] finds that the attached LEV is one of the reasons of high lift for flying animals. The attached LEV induces the higher stall angle and delays the stall of the wing. As the nano rotor blade size is comparable to that of flying insect, the bio-inspired blade motion can be introduced to improve the aerodynamic performance of nano rotor. The blade pitch motion can be employed along the longitudinal axis of blade to keep the leading edge vortex.

As the weight of NAV is limited, it is required that the nano rotor is light and thin enough [13]. Carbon composite laminate is used to fabricate the rotor blade. Because the composite blade is thin and light, the blade is flexible [14]. Due to the unsteady aerodynamic force, centrifugal force and the gravity, the flexible blade suffers the deformation and vibration which influences the flow field of nano rotor. The coupling between structure and fluid has an effect on the propulsive performance of the bio-inspired rotor. Therefore, the blade cannot be treated as a rigid body but a flexible one. The fluid-structure interaction shall be taken into account when studying the propulsive performance of the bio-inspired nano rotor. The research on the nano rotor with FSI method is scarcely reported. But the full-scale rotor is well studied based on FSI using experiment or numerical method in recent years [15-19]. When we study the propulsive performance of the bio-inspired nano rotor with FSI method, the first step is to obtain an accurate finite-element structural model of composite nano rotor. However, the fabrication error and the difference of material parameters will introduce nondeterminacy in the structural model. The modal experiment is usually carried out to validate the structural model. Yang [20], Luo [21] and Mohammad [22] adopted hammer to excite rotors and measured structural response by means of acceleration sensors onto structure surface. However, because traditional contact modal test methods will introduce additional mass and change the boundary conditions, they are not suitable for thin, small and flexible nano rotor. Non-contact modal test methods, for instance acoustic excitation and laser vibrometer, are widely used by simple and large size structures [23, 24]. So, a non-contact modal test method is necessary. Since there are three different motions for the bio-inspired rotor, i.e. the rotation around the hub, the bio-inspired pitch motion, and the deformation due to the aerodynamic force, centrifugal force and the gravity force, the method to describe the coupling motions is another important issue for the study of the bio-inspired rotor. Sliding mesh method and Multiple References Frame method (MRF) are widely used to study the rotating rotor. Sliding mesh method which is suitable for the research on the fine flow field of rotor requires more computational resource than MRF. As we focus more on the propulsive performance of the bio-inspired nano rotor in this research, MRF is used to describe both the rotation motion and the bio-inspired motion. The deformation of the structure is related to local change of solid surface, so the sliding mesh method and MRF fail to describe it. Therefore, the deforming grid method is used. To study the bio-inspired rotor with the FSI method, a weak coupling method is used by transfer data at the interface of the structural model and the fluid model for the bio-inspired nano rotor. The interpolation method for FSI during the simulation is important [25-26]. RBF method constructs a Radial Basis Function and uses it to obtain the unknown parameters at the interface. It is simple and can be used for complex mesh. Therefore, RBF interpolation method is a useful method for this study.

In summary, the bio-inspired unsteady mechanisms are mainly used on flapping wing NAV and scarcely on rotary-wing NAV. And the propulsive performance of the bio-inspired nano rotor is scarcely studied with FSI method. In this study, the bio-inspired unsteady mechanism is introduced to improve the aerodynamic performance of the nano rotor. A non-contact modal test experimental platform is built based on sound excitation instrument and laser vibrometer and a modal test of the nano rotor is carried out. Successively, the finite element model of the nano rotor is established and verified with the modal test. Then propulsive performance of the bio-inspired nano rotor is analysed at different bio-inspired pitch frequency and the response of the blade structure is also analysed with the fluid-structure interaction method.

2 COMPUTATIONAL METHODOLOGIES AND EXPERIMENTAL PLATFORM

2.1 Governing Equations

The blade tip velocity of the nano rotor is lower than 0.1 Mach. The low-speed performance is extremely poor for compressible NS equations because of stiffness of governing equations which is caused by the small ratio of the convective speed to the speed of sound. Therefore, the preconditioning techniques are introduced to eliminate the disparity between the particle and acoustic wave speeds at low speed. The preconditioned governing equations can be rewritten as follows [27].

$$\begin{aligned}\Gamma\frac{\partial}{\partial t}\iiint_V q\,dV &+ \oiint_{\partial V}(E-E_V-u_gQ)\cdot n_x dS \\ &+ \oiint_{\partial V}(F-F_V-v_gQ)\cdot n_y dS \quad (1) \\ &+ \oiint_{\partial V}(G-G_V-w_gQ)\cdot n_z dS = 0\end{aligned}$$

Here, Γ is the preconditioning matrix, Q is vector of primitive flow variables. $\vec{F}$ and $\vec{F}_v$ termed vector of convective fluxes are related to the convective transport of quantities in the fluid. $\vec{F}_v$ termed vector of viscous fluxes contain the viscous stresses τ_{ij}. In the formula, $\vec{U}$ and $\vec{U}_g$ are the velocity component and moving grid velocity component. The equations were solved with finite volume method and Roe's flux scheme was employed.

2.2 Structural Dynamic Equations

The finite element method (FEM) formulations can be established on the basis of the finite deformation theory. Assumed that the composite material of nano rotor is linearly elastic and orthotropic, By taking into account the three loads, corresponding kinetic equation of nano rotor can be written as follows [28].

$$[\boldsymbol{M}]\{\ddot{\boldsymbol{x}}\}+[\boldsymbol{C}]\{\dot{\boldsymbol{x}}\}+[\boldsymbol{K}]\{\boldsymbol{x}\}= \{\boldsymbol{F}_{ce}\}+\{\boldsymbol{F}_{g}\}+\{\boldsymbol{F}_{h}\} \tag{2}$$

where $\ddot{x}$ represents node acceleration, x is node displacement, M is mass matrix, C is damping matrix, K is stiffness matrix ; F_{ce}, F_g and F_h are centrifugal force, gravity and aerodynamic force, respectively.

2.3 Fluids-Structure Interaction Method

A loose-coupling method is used in the FSI simulation. Because the FEM mesh is different from the CFD mesh, the data shall be transferred at the interface. The RBF method is used in this study. The deforming grid method is used in the CFD solver. Time-marching method is a sequential coupling method. The aerodynamic force is firstly calculated with CFD solver. Then the aerodynamic force together with gravity and centrifugal force will be interpolated on the FEM mesh with which the structural dynamic response can be calculated with the FEM solver. The node displacement will be transferred to the CFD solver and the mesh will be updated with displacement. Then the flow field will be calculated again and the aerodynamic force will be obtained. The solver will keep on repeat the above calculation.

2.4 Bio-inspired Pitch Motion

Figure 1 shows a schematic of the motion for the nano rotor. Two blades rotate around the central axis and pitch around the 1/4 chord along the blade. To describe the motion of the nano rotor, two coordinate systems are introduced. The fist coordinate system $O-XYZ$ is an inertial coordinate system which keeps motionless and the other coordinate system $O'-X'Y'Z'$ is a comoving coordinate which rotates with the rotor blade. The origin of the coordinate systems locate on the centre of the 1/4 chord along the blade. The pitch angle, the flapping angle and the roll angle are defined as θ , ϕ and γ . At the beginning, the $O'-X'Y'Z'$ concides with $O-XYZ$. But $O'-X'Y'Z'$ changes with the rotation and pitch for the blade. The transform matrix between the $O-XYZ$ and $O'-X'Y'Z'$ can be write as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -\sin\theta\cos\gamma & \sin\theta\sin\gamma & \cos\theta \\ \cos\theta\cos\gamma & -\cos\theta\sin\gamma & \sin\theta \\ \sin\gamma & \cos\gamma & -\sin\theta \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \tag{3}$$

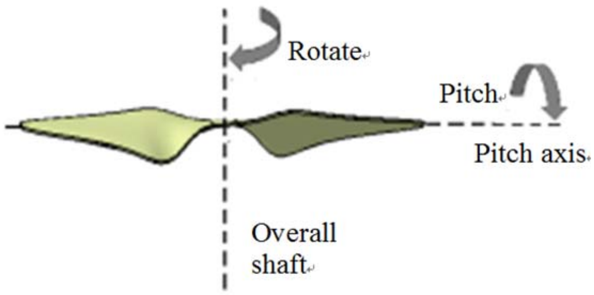


Figure 1 - Schematic of rotor pitch motion.

When inserting a table, you can choose the appropriate style - Table 1 below is an example. Put the caption under the table.

The rotational speed of the nano rotor is 6500RPM. Then the pitch motion of the blade is defined as a sine function

$$\theta(t) = \theta_0 \cdot \sin(2\pi f_p t) + \theta_{hias} \tag{4}$$

where $\theta(t)$ is the pitch angle varying with the time. θ_0 is the amplitude of the pitch motion which is 5° in this study. f_p is the pitch frequency. θ_{bias} is the initial pitch angle which is zero.

2.5 Modal test Experimental Setup

Traditional contact-type modal test method cannot be adopted because extra mass and stiffness will be introduced. In this study, a non-contact modal experiment platform for nano rotor was established based on sound excitation instrument and laser vibrometer as shown in Figure 2. This test platform include supporter of the nano rotor, sound excitation instrument and laser vibrometer. The full frequency speaker was excited by broadband white noise signal, which was also inputted into the Polytec laser vibrometer system as a reference signal. The SPL of the sound generated is from 100 to 110dB. A two-dimensional modal test was performed for the rotor. Seventy to eighty laser scanning points were set on the surface of nano rotor and their vibration displacements were measured by virtue of Doppler Effect. All the measured signals were processed to filter the useless signals and reduce the noise so as to ultimately obtain the accurate frequency spectrum and vibration modes of nano rotor.

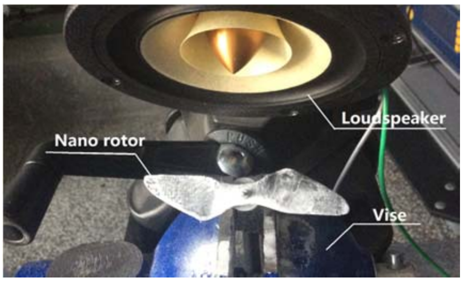


Figure 2 - Modal test bench for nano rotor.

3 RESULTS AND DISCUSSION

3.1 Dynamic Characteristics Analysis of Nano Rotor

The nano rotor using for modal test was laminated by 6 layers unidirectional fibre composite and the stacking sequence in Fig. 4 (a) was [0/90/0/90/0/90]. Every layer thickness was 0.85mm and the total thickness of rotor was 0.51mm. The platform adopted the method of acoustic excitation and laser vibration measurement technology. Five natural frequencies and vibration modes were obtained.

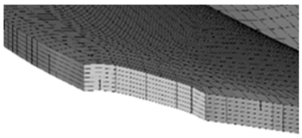


Figure 3 - FE model of nano rotor.

When comparing the experimental results and computational results, it is found that the two groups of vibration modes are extremely similar in flapping, bending and torsion according to the similar vibration modes. The corresponding natural frequencies are also compared in Table 1. It is found that the relative errors for them are lower than 4%. The first two natural frequencies are also comparable to the rotating frequency of the nano rotor, which shall be paid more attention to during the design of the NAV to avoid the resonance. Because the maximum relative error of simulation is lower than 4% and the minimum relative error is even lower than 1%, the finite element model is thought to be capable of reflecting the main structural feature of nano rotor.

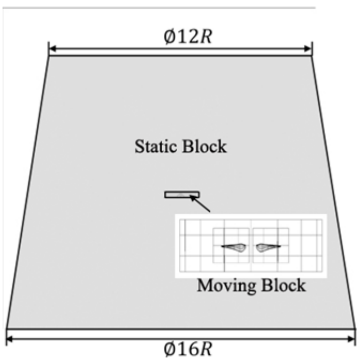
Order	Natural frequency of experiment /Hz	Natural frequency of simulation /Hz	Relative Error/%
1	303.10	294.45	2.84
2	310.90	304.77	1.96
3	1131.30	1176.1	3.96
4	2743.70	2816.8	2.66
5	3181.30	3202.4	0.6

Table 1 - Comparison of natural frequencies between experiment and computation.

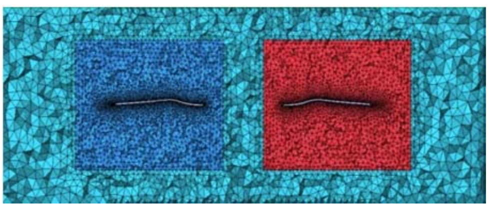
3.2 Propulsive performance of bio-inspired Rigid Nano Rotor

MRF is used to describe the rotation and the pitch motion in this study. The flow field is composed of a static block, a rotation block and a pitch block as shown in Figure 4. The pitch block which contains two blades is a cylinder with a radius of 0.5 R and a height of 1.2R. The rotation block located outside of the pitch block is also a cylinder with a radius of 2.4R and a height of 1.6R. The static block is a cone with a upper radius of 6R and a lower radius of 8R. The structure mesh is is used for static block and the unstructured mesh is used for both the rotation block and the pitch block. The total number of the mesh is about 8 million.

The rotational speed of nano rotor is 6500 RPM. Three cases are calculated including case 1 in which there is no pitch motion, case 2 in which the pitch frequency $f_p = f_0$ and the pitch case 3 with frequency $f_p = 2f_0$. The simulation is carried out on a HP station with 64GB RM and 40 cores.



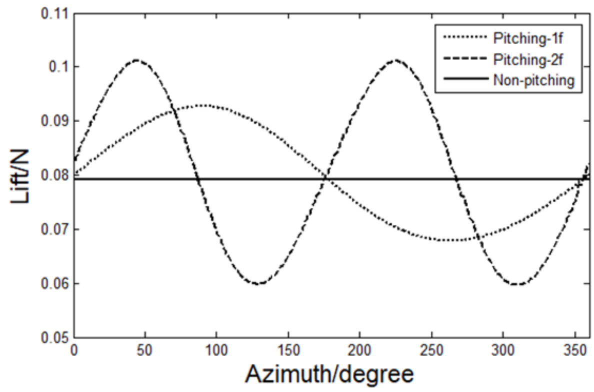
(a) Moving block and static block.



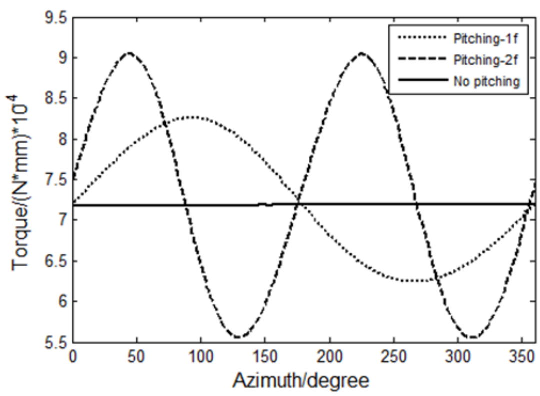
(b) Mesh section.

Figure 4 - Computation Zone and mesh of rotor for CFD.

Figure 5 shows the thrust and the torque of the nano rotor varying with the azimuth. It is found that the curves of thrust and the torque resemble sine or cosine function. And the periods of the curves corresponds with that of the pitching motion. The frequency of the curve increases with the pitching frequency. The amplitude of the thrust and torque increases with the frequency as well. For the non-pitching case, the value of the thrust and the torque keeps as an equilibrium value.



(a) Lift.



(b) Torque

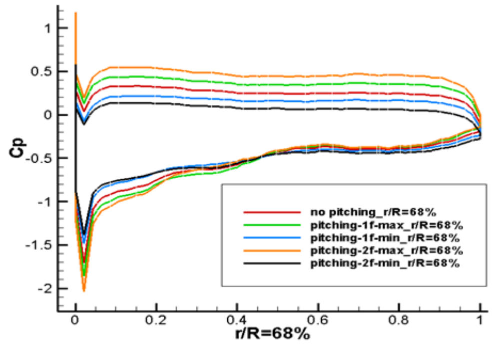
Figure 5 - Lift and torque curves of pitching rigid rotor in one cycle

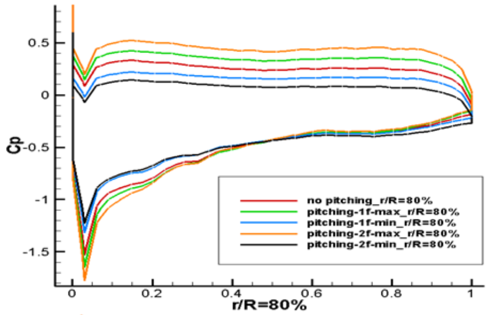
The average value in a cycle for the thrust coefficient, the torque coefficient and the figure of merit are summarized in table 2. It is found that the thrust coefficient increase with the pitching frequency. The non-pitching case obtains the minimum FM. And the maximum FM is achieved by the pitching case with high pitching frequency. It is indicated that the propulsive performance of the nano rotor is improved with the bio-inspired pitching motion.

Cases	Thrust coefficient	Torque coefficient	FM
Case 1	0.0450	0.0109	0.625
Case 2	0.0454	0.0108	0.633
Case 3	0.0455	0.0108	0.635

Table 2 – Propulsive performance for different cases.

Figure 6 shows the pressure coefficient varying with chord at the sections of r/R=68%, 80% and 96% along the blade for the three case. It is found that the difference of the pressure coefficient at the suction surface and press surface increases with the pitching angle and pitching frequency. At the high pitching angle, the pressure coefficient at the press surface for case 2 and case 3 is higher than that for case 1. The variation of the pressure coefficient with the pitching angle indicates that the pitching motion changes the velocity of the LEV shedding. At high pitching angle, LEV attaches on the surface which induces high pressure coefficient difference. However, the pressure coefficient curve varies from the other at the section of r/R=96% due to the blade tip vortex.





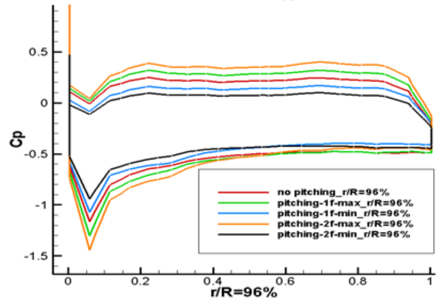


Figure 6 - Pressure coefficient comparison at different blade sections for different cases.

Figure 7 shows iso-surface of the magnitude of vorticity. At high pitching angle, the biggest iso-surface is shown which indicates that the vorticity is the strongest. The pitching motion influences the vorticity and propulsive performance of the nano rotor as a result.

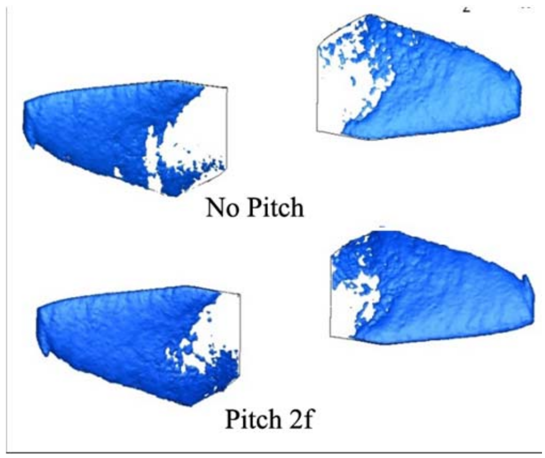
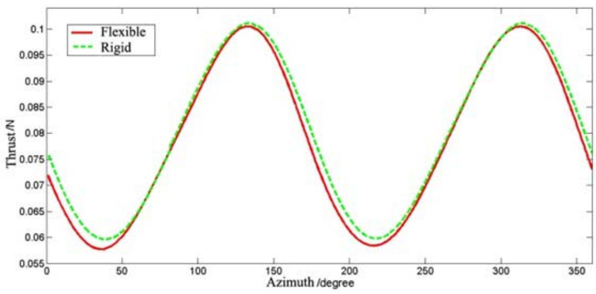


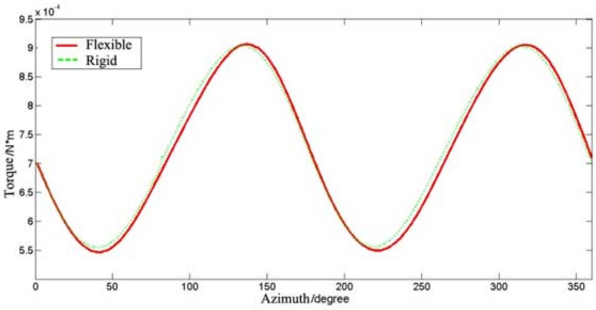
Figure 7 – Iso-surface of vorticity for the non-pitch nano rotor and the pitching nano rotor.

3.3 Propulsive performance of bio-inspired flexible Nano Rotor

The bio-inspired flexible nano rotor with pitching frequency of $f_p = 2f_0$ is also studied with FSI method. Figure 8 shows the thrust and the e torque varying with azimuth in one cycle for both flexible rotor and rigid rotor. It is found that the thrust of the flexible nano rotor is slightly lower than that of the rigid one. However, the torque of the flexible nano rotor approximates that of the rigid one. The average thrust coefficient and torque coefficient of both flexible Nano rotor and rigid nano rotor are also shown in Table 3. Results show that both the thrust coefficient and torque coefficient are lower than that of the rigid nano rotor. As the more drop can be found in the thrust coefficient, the FM of flexible is lower than that of rigid one. It is indicated that the propulsive performance of nano rotor degrades with the flexibility of nano rotor and it is necessary to carry out the FSI method when we investigate the propulsive performance of the nano rotor.



(a)



(b)

Figure 8 – Thrust and torque varying with the azimuth for both flexible and rigid nano rotor.

Cases	Thrust coefficient	Torque coefficient	FM
Flexible	0.0446	0.0106	0.628
Rigid	0.0455	0.0108	0.635

Table 3 – Propulsive performance for flexible and rigid nano rotor.

Figure 9 shows the pressure coefficient at different blade stations i.e. r/R=50% and 96% for both flexible and rigid bio-inspired nano rotor when the rotor achieves the maximum thrust coefficient. It is found that the curve of the pressure coefficient for the flexible nano rotor nearly locates inside of that for the rigid nano rotor at all the blade stations except at r/R=96%. The deformation of the nano rotor at the blade tip is large which might induces the difference.

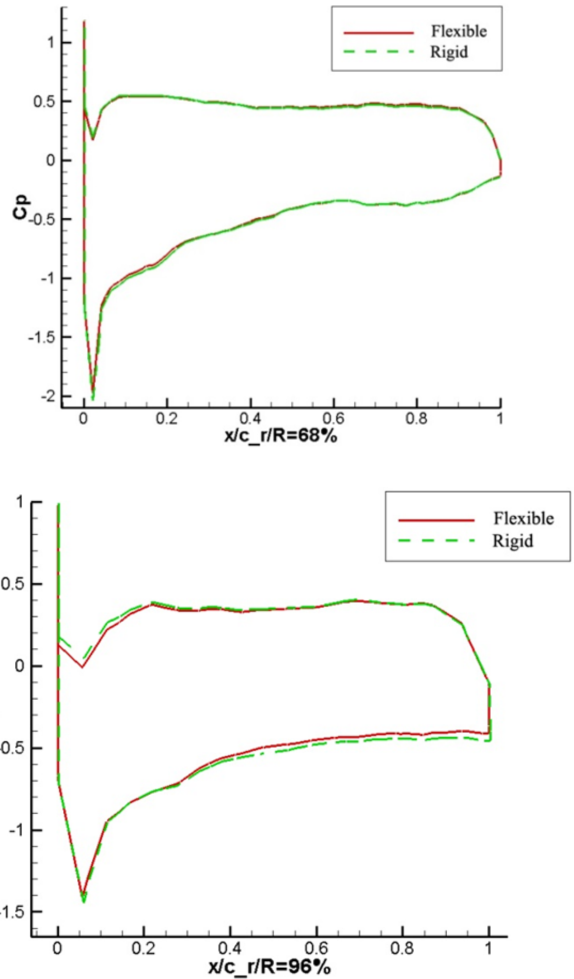
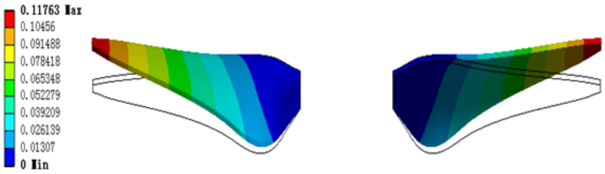


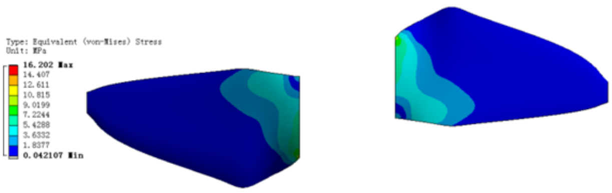
Figure 9 – Thrust and torque varying with the azimuth for both flexible and rigid nano rotor.

3.4 Structural response of bio-inspired flexible Nano Rotor

Figure 10 shows Displacement and stress contour of the nano rotor. At the beginning, the maximum displacement vibrates irregularly. The maximum amplitude is as high as 0.2 mm. Then, it begins to vibrate regularly. Results show that the maximum of the displacement is about 0.118mm at the blade tip. The maximum stress is found at the blade root. The two blades tilt due to the aerodynamic force.



(a) Displacement Contour



(b) Stress contour

Figure 10 - Displacement and stress contour of the nano rotor.

The amplitude spectrum of displacement at the blade tip is shown in Figure 11. There are two peak values at frequency of 211.9 Hz and 605.46Hz. When comparing with the modal test results, it is found that the first frequency is close to the rotational frequency. And the second frequency is close to the fundamental frequency of the nano rotor.

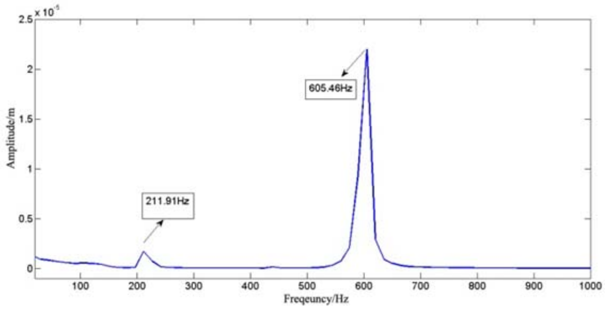


Figure 11 – Amplitude spectrum of displacement

6 CONCLUSION

In this paper, the bio-inspired composite nano rotor is studied with FSI method. Blade pitch motion is introduced to improve the propulsive performance of the nano rotor performance. The bio-inspired blade motion is introduced to improve the propulsive performance of nano rotor at an ultra-low Reynolds number. A non-contact modal experimental platform is firstly established using sound excitation instrument and laser vibrometer. The structural characteristics of the composite nano rotor are measured. It is found that the natural frequencies are very close for the first two orders. The finite element model of composite rotor is created accordingly. The modal is studied computationally with FEM solver. It is found that the simulation results matched well with the experimental results which verified the correctness of the finite element model. The CFD model is established and the propulsive performance of the rigid bio-inspired nano rotor motion is studied at different pitch frequencies. Results show that the thrust and torque for the bio-inspired pitching rotor are higher than those for the rotor without bio-inspired motion and the propulsive performance for the nano rotor with bio-inspired pitching frequency of two times of that rotation frequency is higher than that with only one times pitching frequency. It is evident that the improvement enhanced with the increase of the pitching frequency. The flexible bio-inspired nano rotor is then investigated with FSI method. The results show that the propulsive performance of the flexible nano rotor is lower than that of the rigid nano rotor. It is evident that it is necessary to consider the flexibility of the composite nano rotor when investigating the propulsion performance of bio-inspired nano rotor. Then the response of blade structure is also analysed. Results show that the blade structure vibrates at small amplitude. In general, it is found that the bio-inspired pitching motion can improve the performance of nano rotor. In the future, the experiment will carry out to further verify the c

ACKNOWLEDGEMENTS

Authors are grateful for the support from the National Natural Science Foundation of China through grant numbers 11772252 and 11302164. The authors also acknowledge the support of K. C. Wong Education Foundation.

REFERENCES

[1] 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.

[2] S. Nolfi. Power and the limits of reactive agents. Neurocomputing, 42(1–4):119–145, 2002.

[3] C.M. Bishop. Pattern recognition and machine learning. Springer Science and Business Media, LLC, New York, NY, 2006.

[1] H. Todd, ,M. Christopher, and T. Richmon, The DARPA Nano Air Vehicle Program. In 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2012.

[2] S. Sato, M. Drela, J.H. Lang, and D.M. Otten. Design and characterization of hover nano air vehicle propulsion system. In 27th AIAA Applied Aerodynamics Conference, 2009.

[3] H. Youngren, C. Kroninger, M. Chang, and S. Jameson. Low Reynolds number testing of the AG38 airfoil for the SAMARAI nano air vehicle. In 46th AIAA Aerospace Sciences Meeting and Exhibit, 2008.

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

[5] Z. Liu, J.M. Moschetta, N. Chapman, R. Barenes, and M. Xu. Design of Test Benches for the Hovering Performance of Nano-Rotors. International Journal of Micro Air Vehicles, vol. 2(1):117-32, 2010.

[6] R. J. Wootton. The Mechanical Design of Insect Wings. Scientific American, 263:114-120, 1990.

[7] N. Phillips, K. Knowles. Effect of flapping kinematics on the mean lift of an insect-like flapping wing. Proceedings of the Institution of Mechanical Engineers Part G Journal of Aerospace Engineering, 225:723-736, 2011.

[8] W. Shyy, H. Aono, S. K. Chimakurthi, et al. Recent progress in flapping wing aerodynamics and aeroelasticity. Progress in Aerospace Sciences, 46(7):284-327, 2010.

[9] S. Guo, D. Li, J. Wu. Theoretical and experimental study of a piezoelectric flapping wing rotor for micro aerial vehicle. Aerospace Science & Technology, 23(1):429-438, 2012.

[10]B. Fitchett Development and Investigation of a Flapping Rotor for Micro Air Vehicles. Dissertations & Theses - Gradworks, 2007.

[11] N.A. Koratkar, and I. Chopra. Open-loop hover testing of a smart rotor model. AIAA Journal, Vol. 40(8): 1495-1502, 2002.

[12] C.P. Ellington. The novel aerodynamics of insect flight: applications to micro-air vehicles. Journal of Experimental Biology, 202(23):3439-3448, 1999.

[13] N. Tsuzuki, S. Sato, and T. Abe. Design guidelines of rotary wings in hover for insect-scale Micro Air Vehicle applications. Journal of Aircraft, vol. 44(1): 252-263, 2002.

[14] Z. Liu. Design and aero-propulsive analysis of a nano air vehicle. Ph. D., L'INSTITUT SUPÉRIEUR DE L'AÉRONAUTIQUE ET DE L'ESPACE, Toulouse, 2011.

[15] F. Bohorquez, D.J.Pines, and P.D. Samuel. Small rotor design optimization using blade element momentum theory and hover tests. Journal of Aircraft, vol. 47(1): 268-283, 2010.

[16] A. Smedresman, D. Yeo, and W. Shyy. Design, fabrication, analysis, and testing of a micro air vehicle propeller. In 29th AIAA Applied Aeroydnamics Conference, 2011.

[17] P. Singh, and C. Venkatesan, , Experimental Performance Evaluation of Coaxial Rotors for a Micro Aerial Vehicle, Journal of Aircraft, vol. 50(5): 1465-1480, 2013,.

[18] M. I. Kellogg, and B. W. J. Parametric design study of the thickness of airfoils at Reynolds numbers from 60,000 to 150,000. In 42nd AIAA Aerospace Sciences Meeting and Exhibit, 2004.

[19] D. Schafroth, S.Bouabdallah, C.Bermes, and et, al. From the test benches to the first prototype of the muFly micro helicopter. Journal of Intelligent & Robotic System, vol. 54 (1): 245-260, 2009.

[20] Y. T. Yang, C. Q.Bai, H. Y. Zhang, et al. Study on Helicopter Rotor's Dynamic Characteristics in a Combining Environment. Chinese Journal of Applied Mechanics, 03: 446-451, 2014.

[21] Z. D. Luo. Research on Vibration of Engine Cool Fan Based on Air-structure Coupling. South China University of Technology, 2012.

[22] H. J. Mohammad, G. Mostafa, Z. R. Saeed, and S. Behrooz. Dynamic Analysis of a High Speed Rotor-bearing System. Measurement, 53: 1–9, 2014.

[23] L. Li, Q. W. Guo, H. W.Song, et al. Experimental Modal Analysis and Parameter Identification of an Aluminum Plate Excited by Sound. Noise and Vibration Control, 06: 105-110. 2009.

[24] Y. F. Xu, and W. D. Zhu. Operational Modal Analysis of a Rectangular Plate Using Non-contact Excitation and Measurement. Journal of Sound and Vibration, 332: 4927-4939, 2013.

[25] R. M. V. PIDAPARTI. Structural and aerodynamic data transformation using inverse isoparametric mapping. Journal of Aircraft, 29(3): 507-509, 1992.

[26] C. B. Allen, and T. C. S. Rendall. Unified Approach to CFD-CSD Interpolation and Mesh Motion using Radial Basis Functions. In 25th AIAA Applied Aerodynamics Conference, 2007.

[27] J. M. Weiss, and W. A. Smith. Preconditioning applied to variable and constant density flows. AIAA Journal, 33: 2050-2057, 1995.

[28] Z. N. Zhou, Y. M. Li, Y. X. Gu, et al. Dynamic Characteristic Analysis of Blade Based on Fluid-Structure Coupling. Journal of China University of Mining&Technology, 38(3):120-126, 2009.

Force-directed formation and guidance for large scale swarming system of micro air vehicles

Longjiang Li*, Hongmei Zhou, Jianjun Yang, and Yuming Mao
School of Information and
Communication Engineering,
University of Electronic Science and
Technology of China,
Chengdu, 611731 China

ABSTRACT

This paper proposes a force-directed method for formation and guidance of large scale swarming system, consisting of hundreds or even thousands of MAVs, in which two independent communication channels are configured, so that local repulsion and global aggregation can be coordinated to control the behavior of the whole swarm. Local repulsion force is committed to ensure all MAVs stay within a safe distance of each other as the swarm constantly adjusts, while global aggregation force is employed for guiding the whole swarm. By arranging multiple leaders in proper way, swarm guiding, reshaping, partitioning and merging can be supported in a consistent way. Besides, the calculation of local repulsion force only depends on relative location and angle information, so that the safety of the swarm can be enforced even without depending on the precisely positioning information, which is extremely important for the applicability of the proposed scheme in real environments. Finally, the simulation demonstrates that local repulsion and global aggregation may work together well, leading to a stable formation of MAVs, with enough freedom of control guided by leaders.

1 INTRODUCTION

Due to its potential applications in battlefields and disaster scenes[1], Micro Air Vehicles (MAVs), as a class of miniature unmanned aerial vehicles (UAVs) that have size restriction, have gained many attentions from industry and academic[2]. For single MAV, the mission abilities are limited, such as for reconnaissance and surveillance. In order to increase mission success rates and enlarge field of view, hundreds or even thousands of low-cost MAVs fly as a flock to replace an expensive multifunction UAV, and achieve a desired formation pattern to meet various task's requirements, which is called formation flight control of MAVs[3].

At present, the methods of formation control can be classified into leader-follower method [4][5], virtual structure method [6], artificial potential field based method[7] and consensus-based method [8][9]. Leader-follower method refers to the control mechanism where one or several MAVs serve as leaders, and the rest of MAVs follow the leaders' trajectory according to a certain formation to perform a series of required tasks, but most of these methods only consider a small scale of swarms, at most with dozens of vehicles[10][11]. The idea of virtual structure[12] is to regard the multi-agent formation as a rigid body, but it is usually challenging to maintain the virtual structure in a consistent way, which limits the scalability of the swarm[6]. Artificial potential function-based method adopts a mixture of attractive and repulsive potential field for guiding the swarm[13], which is widely used to solve obstacle avoidance problems in a complex environment. In contrast, consensus-based method is abstracted from the swarming behaviors of animals[14][15][16] and most researches formulate the swarm as a consistency problem [17], but most local operations, which depend on multi-hop packet transfers, have a remarkable control delay and do not allow for too much freedom of control and guidance[18].

Moreover, most of existing formation models depend on accurately positioning information, which is hard to obtain through traditional communication channels in real environments due to the problem of bandwidth constraints, interference and noisy, especially when a large number of MAVs are involved with high mobility. [19] used a monocular on-board camera for detecting and estimating the pose of micro aerial vehicles. [20] proposed a vision-based tracking scheme to achieve the formation tracking, which uses a Kinect installed on each agent as the sole sensor to get UAV's status, but there are problems of visibility constraints because of the limited vision range of Kinect.

This paper proposes a force-directed method for formation and guidance of large scale swarming system, maybe consisting of hundreds or even thousands of MAVs. The basic idea is to mandate the safety distance between neighbors through local repulsion force, while global aggregation force is employed for guiding the behavior of the whole swarm with enough freedom. Two independent communication channels

*Email address(es): longjiangli@uestc.edu.cn

are configured for each MAV, so that the formation of the swarm can be constructed through the balance of local repulsion and global aggregation. Since it is not easy to obtain accurately positioning information, the calculation of local repulsion only uses the information of relative position and relative angle of neighbors, which greatly improves the feasibility of the proposed scheme in the real application's environments.

2 System Model

2.1 *Problem description*

Compared with UAV, Micro Air Vehicle (MAV) has a smaller size but the swarm size of MAVs is usually much bigger, even up to hundreds or thousands, which makes it rather challenging to keep formation and guidance for such a swarming system of MAVs.

We assume that all MAVs have the same hardware configuration, in which each MAV has two independent communication channels. One channel works in the short-range mode, configured for supporting for information exchange between neighboring MAVs, so that local repulsion can be committed to ensure that all MAVs stay within a safe distance of each other as the swarm constantly adjusts. The other channel works in the long-range mode, which may cover the range of whole swarm. By using the long-range channel to broadcast the location and speed of leaders to every other MAVs, each MAV will predict the state of leaders and apply a global aggregation force to itself, so that leaders can be followed in an autonomous way.

Thus, the key point of the proposed scheme is to make formation by balancing the local repulsion and the global aggregation, and, at the same time, to allow the swarm of MAVs to have enough freedom for easily being guided through commands.

2.2 *Formation of MAV swarm*

Each MAV calculates the repulsion force from its neighbours according to the neighbors' state information obtained through the short-range channel, and calculates the global attraction force from its leader according to the leader's state information obtained through the remote channel. By balancing the repulsion force and the attraction force to control each MAV state, a stable UAV formation centered at the leader can be established, as shown in Fig. 1.

2.2.1 Repulsion force

A repulsion force is employed to ensure a safe distance between every two of MAVs, so that collisions can be avoided. Through the short-range channel, a MAV, i, is able to obtain the state information of its neighbors within the communication range, so that a repulsion force, $F_{i,j}^{(R)}$, can be calculated for each of the neighbors, as follows.

$$F_{i,j}^{(R)} = -F_R \times x_{i,j}^{\alpha} * \vec{d}_{i,j} \tag{1}$$

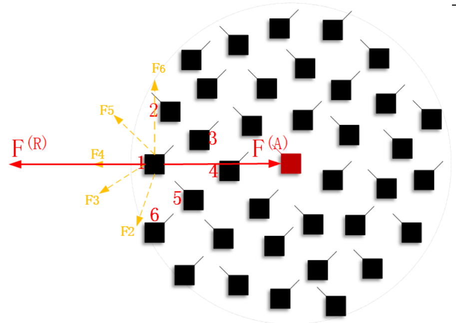


Figure 1: Balanced force for formation of MAVs

where $x_{i,j}$ is the distance between i and j, $\vec{d}_{i,j}$ the unit vector indicating the direction from i to j, and F_R the fixed repulsion force that can be pre-configured, and α the repulsion coefficient for tuning the effects of the distance. In case that $\alpha = 2$, Eqn.(1) indeed follows the Coulomb's inverse-square law, analogous to Isaac Newton's inverse-square law of universal gravitation.

2.2.2 Aggregation force

In order to aggregate MAVs into a swarm, an aggregation force is also needed. We assume that one or more MAVs play the role of leaders, which keep broadcasting their own state information periodically to all other MAVs through the long-distance channel. Based on the state information of leaders, each MAV calculates the attraction force from its leader.

$$F_i^{(A)} = F_A \times x_i^{\beta} * \vec{d}_i \tag{2}$$

where x_i denotes the distance between MAV i and the leader, $\vec{d}_i$ the unit vector from MAV i to the leader, F_A the fixed attraction force, and β the attraction coefficient, for tuning the effects of the distance.

2.2.3 Convergence of resultant force

By considering the total effects of repulsion force and aggregation force, each MAV, i, controls its kinematic state based on the resultant force, F_i, as follows.

$$F_i = F_i^{(A)} + \sum_{j \in V_i} F_{i,j}^{(R)} \tag{3}$$

where V_i indicates the set of neighboring MAVs that are located in the communication range of i. When F_i is 0, a balance state is reached. Otherwise, the repulsion force dominates the kinematic state of i when $F_i < 0$, or the aggregation force dominates. When almost every MAV falls into the balance state, i.e., $F_i = 0$, we say that the whole swarm reaches a equilibrium state.

2.3 Guidance of MAV swarm

In real application's environments, the formation of MAVs is usually needed to be adjusted frequently for meeting the requirements of various tasks.

2.3.1 Swarm guiding

Since each MAV is attracted by the aggregation force of its leader, the swarm may be guided by controlling the state of the leader, so that the swarm follows the flight trajectory of the leader. By collecting the state information of leaders through the short-range channel, each MAV may predict the moving direction of its leader. When the distance that the leader moves is little, the swarm can fly smoothly. Otherwise, if the distance changes too fast, e.g., x_i exceeds a specified threshold, X, an interpolation is usually useful for reducing or avoiding the turbulence of the swarm. As shown in Fig. 2, the coordinate position of the leader is "1" at the previous moment, and it is "4" at this time. In case that the distance between "1" and "4" is very large, linear interpolation is performed on "1" and "4" to smooth the transition by inserting two intermediate points "2" and "3". Then, each MAV will update the location of the leader from "1" and "4" in a specified rhythm, so that a smooth guidance of the swarm can be achieved.

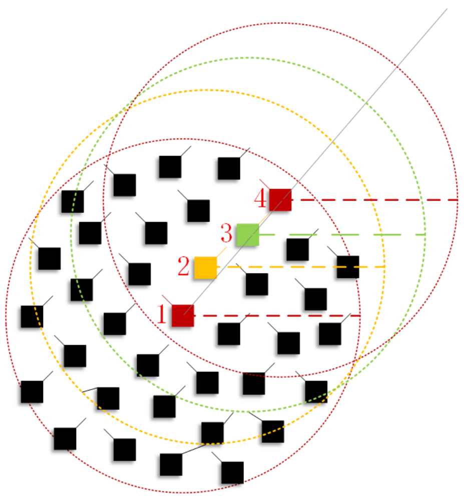


Figure 2: Guidance of MAV swarm

2.3.2 Swarm reshaping

When only one leader is employed, the shape of the swarm always tends to be like a circle in two-dimension space or a ball in three-dimension space. If a different shape is needed, one of the easiest ways is to use multiple leaders, so that the shape of the swarm may be adjusted by controlling the locations of these leaders. As shown in Fig. 3, by assigning multiple leaders, a specified flight path may be achieved for the swarm.

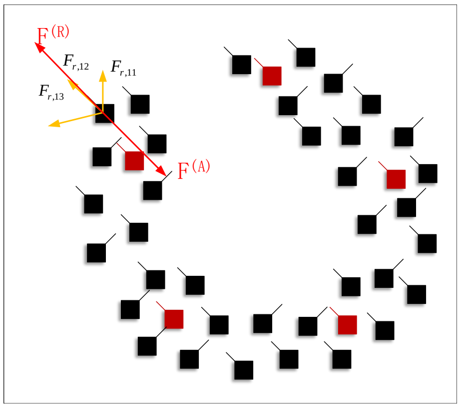


Figure 3: Reshaping of MAV swarm

2.3.3 Swarm partitioning and merging

Due to various requirements of applications, if the number of leaders changes, swarm partitioning or merging are usually involved, as shown in Fig. 4.

Swarm partitioning may take place when the number of leaders increases. When the number of leaders is more than one, a membership between MAVs and leaders must be specified, so that each MAV knows which leader is what it should follow. In this case, all leaders periodically broadcast not only their state information but also the membership information, so that the swarm may be partitioned by reassigning the membership.

Note that the membership only influences the calculation of aggregation forces, while the calculation of repulsion forces stays the same for every two neighboring MAVs, so that safe distance between every two of MAVs can be always maintained. Though multiple leaders may result in interferences on the long-range channel. As the number of leaders is often small, the interference usually can be overcome or avoided by properly configuring the channel resources.

In contrast, when the number of leaders decreases, the merger of swarms will take place. Similarly, membership can be actively adjusted via the long-range channel, and new balance of repulsion force and aggregation force can be rebuilt after swarms complete the merging process. In some cases, due to the failure or the crash of some leaders, some MAVs may fail to receive the broadcast messages from its leader, so that a merger process takes place passively. Usually, it can be treated as a backup mechanism by enabling each MAV to automatically follow a new leader, in case that it fails to receive the broadcast message from its original leader within a certain period of time.

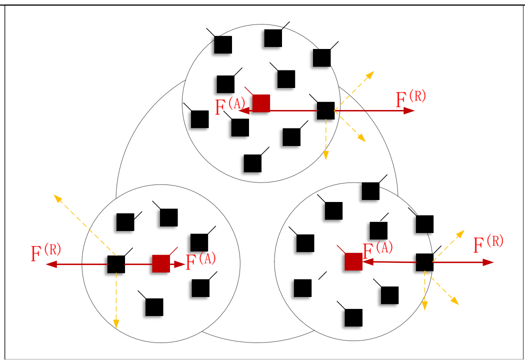


Figure 4: Division and merger of MAV groups

3 Performance Evaluation

In order to investigate the characteristics of the proposed scheme, we implemented both two-dimensional(2D) and three-dimensional(3D) versions parallelly and evaluated the performance by varying parameters.

3.1 Simulation setting

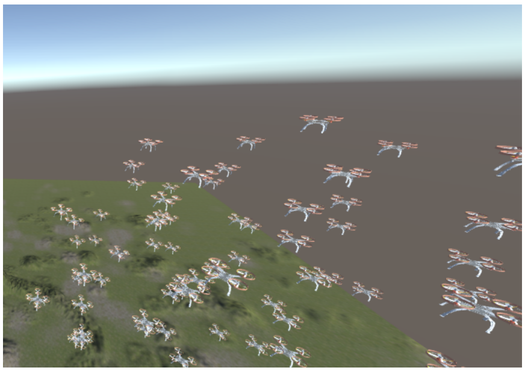


Figure 5: A snapshot of the simulations with 100 MAVs

We implemented a 2D simulation of MAV formation in C# programming language with Microsoft Visual Studio, and a 3D simulation with Unity, which is a 3D game development engine for high-quality 3D interactive supports[21]. The two versions of simulation were implemented by using the same algorithms with the same parameters, and the only difference is that mathematical representations and operations of vectors were performed in a 2D and 3D way, respectively.

In Unity, as shown in Fig. 5, all MAVs are instantiated from the same resource file of a MAV model with different parameters. Each MAV is configured as a rigid body, with parameters, “mass”, “drag” and “angular drag” to be 1, 0.1 and 0.05, respectively. The communication radius is 1000 meter in default. We define a metrics, neighbor distance, for a MAV as the distance that it has from its nearest neighbor as follows.

$$D^{(i)} = \min_{j \in V_i} dist(i,j) \tag{4}$$

where V_i is the set of neighbors for MAV i.

Then, the minimum value of $D^{(i)}$ indicates the safe distance that the swarm has in the equilibrium state.

$$D_{min} = \min_{j \in V} D^{(i)} \tag{5}$$

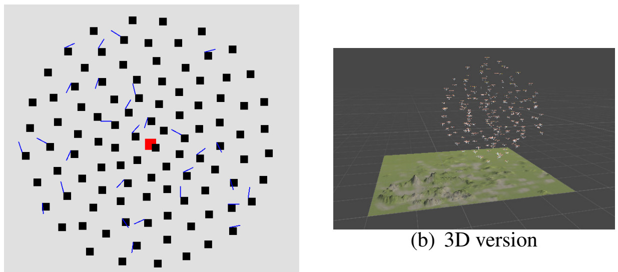
where V is the set of all MAVs.

Correspondingly, the maximum value of $D^{(i)}$ reflects the maximal degree of freedom that a MAV may have in the swarm.

$$D_{max} = \max_{j \in V} D^{(i)} \tag{6}$$

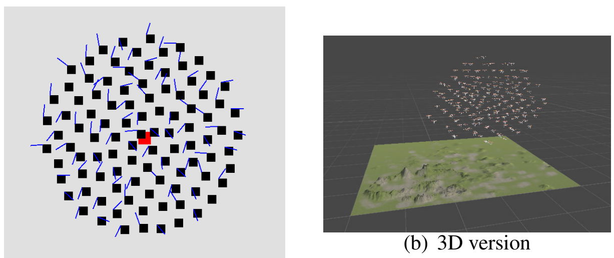
3.2 Results

We firstly compare 2D version and 3D version to validate the correctness of the algorithm implementation. Fig.6 and Fig.7 show the snapshots of the simulation scenario with one hundred MAVs with $F_A = 1000$ and $F_A = 10000$, respectively. From the figure, we can see that Fig.6 has a less intensive distribution of MAVs than Fig.7, because the formation of MAVs is based on the balance of local repulsion force and global aggregation force. Since F_A indicates the fixed repulsion force, a larger F_A leads to a less density of MAVs, which is consistent with Eqn. (1).



(a) 2D version (b) 3D version

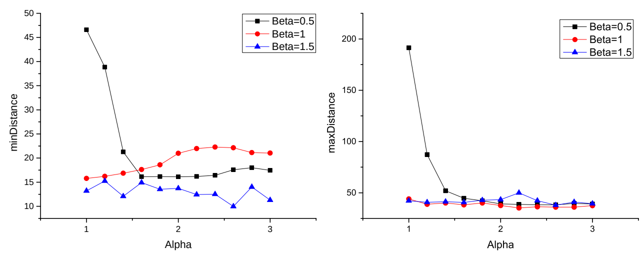
Figure 6: Formation of MAVs with smaller F_A



(a) 2D version (b) 3D version

Figure 7: Formation of MAVs with larger F_A

Fig. 8, Fig. 9 and Fig. 10 show the minimum and maximum of neighbor distance varying with parameters, α, β and the communication range of MAV, respectively. From the picture, we observed that the minimum of neighbor distance is always larger than zero, which means that there is no collision between any two MAVs throughout all simulations.



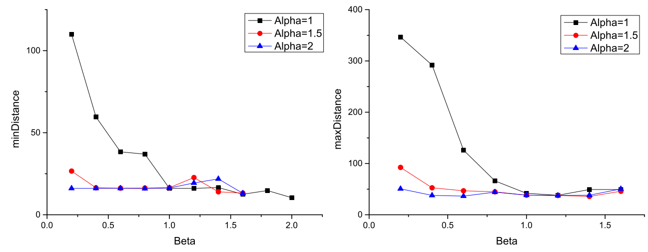
(a) Minimum distance (b) Maximum distance

Figure 8: Neighbor distance varying with alpha

From Fig. 8(a), we can see some interesting results. Firstly, the minimum of neighbor distance first decreases rapidly and then increases slowly as the α increases when $\beta = 0.5$. The decreasing tendency can be explained that a smaller α results in a larger repulsion force. However, since the leader is located at the center of the swarm, when α increases further, the spanning diameter of the whole swarm decreases. Thus, increasing α further results in that the minimum of neighbor distance decreases. Secondly, when $\beta = 1.5$, the minimum of neighbor distance has a large fluctuation with the increasing of α. The aggregation force to a MAV is great when the β is large, but the increase of α will reduce the repulsion force of the MAV. Thus, the fluctuation demonstrates that the balance of repulsion and aggregation is closing to a critical point, on which any small random factors may result in an imbalanced effect. Correspondingly, as shown in Fig. 8(b), the maximum of neighbor distance has similar tendencies, but three curves tend to be close to the same value as α increases, which means that α has little effect on the freedom of swarm. Thirdly, when $\beta = 1$, the minimum of neighbor distance increases slowly with the increase of α, which reflects that the effects of α become less obvious.

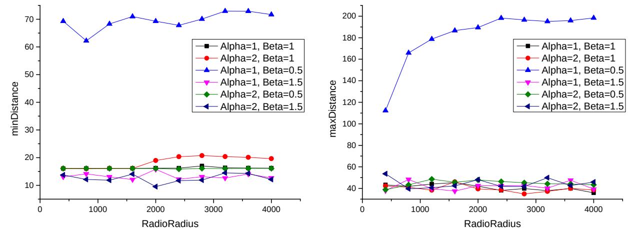
In Fig. 9(a), with the increase of β, the minimum of neighbor distance shows a downward trend, which is steep when α is 1, but not when α is 1.5 and 2. This is because when α is large, the repulsion force between MAVs will be very large, at this time the effect of increasing attraction caused by β is not obvious. Correspondingly, as the α increases, the maximum of neighbor distance in Fig. 9(b) have similar tendencies to those in Fig. 9(a) due to the same reason.

In Fig. 10(a) and Fig. 10(b), when $\alpha = 1$ and $\beta = 0.5$, the minimum and maximum of neighbor distance are both large, which indicates that the MAV formation is dispersed. When α and β are on other combinations, the values of neighbor distance are relatively stable, because there are more MAVs in



(a) Minimum distance (b) Maximum distance

Figure 9: Neighbor distance varying with beta



(a) Minimum distance (b) Maximum distance

Figure 10: Neighbor distance varying with radio radius

the communication range of each MAV, and thus the change of the communication range has a little effect, relatively.

4 Conclusion

In this paper, we present a force-directed method for formation and guidance of large scale swarming system, in which hundreds or even thousands of micro air vehicles are able to construct a swarm with enough freedom of control and guidance. Two independent communication channels are configured, so that local repulsion and global aggregation can be coordinated to control the behavior of the whole swarm. By arranging multiple leaders in proper way, swarm guiding, reshaping, partitioning and merging can be supported in a consistent way. Besides, the calculation of local repulsion force only depends on relative location and angle information, so that the safety of the swarm can be enforced even without depending on the precisely positioning information, which is extremely important for the swarm deployed in the real environments.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61273235, 61601097), the Fundamental Research Funds for the Central Universities of China(No. ZYGX2016J001), the Defense Advance Research Foundation of China under Grants 61401310105 and the Chongqing

Research Program Basic Research Frontier Technology (No. cstc2017jcyjA1246).

References

[1] K. Mase and H. Okada. Message communication system using unmanned aerial vehicles under large-scale disaster environments. In *Proc. and Mobile Radio Communications (PIMRC) 2015 IEEE 26th Annual Int. Symp. Personal, Indoor*, pages 2171–2176, August 2015.

[2] S. Mousavi, F. Afghah, J. D. Ashdown, and K. Turck. Leader-follower based coalition formation in large-scale UAV networks, a quantum evolutionary approach. In *Proc. IEEE INFOCOM 2018 - IEEE Conf. Computer Communications Workshops (INFOCOM WKSHPS)*, pages 882–887, April 2018.

[3] F. Liao, R. Teo, J. L. Wang, X. Dong, F. Lin, and K. Peng. Distributed formation and reconfiguration control of vtol uavs. *IEEE Transactions on Control Systems Technology*, 25(1):270–277, Jan 2017.

[4] G. L. Mariottini, F. Morbidi, D. Prattichizzo, N. Vander Valk, N. Michael, G. Pappas, and K. Daniilidis. Vision-based localization for leadercfollower formation control. *IEEE Transactions on Robotics*, 25(6):1431–1438, 2009.

[5] D. Richert and J. Corts. Optimal leader allocation in UAV formation pairs under no-cost switching. In *Proc. American Control Conf. (ACC)*, pages 3297–3302, June 2012.

[6] X. r. Li, B. Li, P. Zhang, J. y. Zhang, and S. Zhang. Large-scale unmanned aerial vehicle formation maintenance algorithm based on virtual structure. In *Proc. 8th Int. Symp. Computational Intelligence and Design (ISCID)*, volume 1, pages 417–421, December 2015.

[7] N. Zunli, Z. Xuejun, and G. Xiangmin. UAV formation flight based on artificial potential force in 3D environment. In *Proc. 29th Chinese Control And Decision Conf. (CCDC)*, pages 5465–5470, May 2017.

[8] X. Wu, Z. Yang, J. Huo, and J. Zhang. UAV formation control based on consistency. In *Proc. Identification and Control (ICMIC) 2015 7th Int. Conf. Modelling*, pages 1–5, December 2015.

[9] H. Zhang, Y. Li, and G. Xu. Consensus-based control for unmanned aerial vehicle formation with speed adaptability. In *Proc. Chinese Control And Decision Conf. (CCDC)*, pages 6106–6111, June 2018.

[10] W. Yuan, Q. Chen, Z. Hou, and Y. Li. Multi-uavs formation flight control based on leader-follower pattern. In *2017 36th Chinese Control Conference (CCC)*, pages 1276–1281, July 2017.

[11] Y. Jia, Q. Li, and S. Qiu. Distributed leader-follower flight control for large-scale clusters of small unmanned aerial vehicles. *IEEE Access*, 6:32790–32799, 2018.

[12] M. Anthony Lewis and Kar Han Tan. High precision formation control of mobile robots using virtual structures. *Autonomous Robots*, 4(4):387–403, 1997.

[13] Y. Zhao, L. Jiao, R. Zhou, and J. Zhang. UAV formation control with obstacle avoidance using improved artificial potential fields. In *Proc. 36th Chinese Control Conf. (CCC)*, pages 6219–6224, July 2017.

[14] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. *Acm Siggraph Computer Graphics*, 21(4):25–34, 1987.

[15] T Vicsek, A Czirk, E Benjacob, I I Cohen, and O Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226, 1995.

[16] Chenlong He, Zuren Feng, and Zhigang Ren. Collective motion of self-propelled particles without collision and fragmentation. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 003228–003233, Oct 2016.

[17] Z. Lin, L. Wang, Z. Han, and M. Fu. Distributed formation control of multi-agent systems using complex laplacian. *IEEE Transactions on Automatic Control*, 59(7):1765–1777, July 2014.

[18] X. Chen, G. y. Li, and X. m. Chen. Path planning and cooperative control for multiple uavs based on consistency theory and voronoi diagram. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 881–886, May 2017.

[19] M. Zhang, F. Lin, and B. M. Chen. Vision-based detection and pose estimation for formation of micro aerial vehicles. In *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, pages 1473–1478, Dec 2014.

[20] X. Liu, S. S. Ge, and C. H. Goh. Vision-based leader-follower formation control of multiagents with visibility constraints. *IEEE Transactions on Control Systems Technology*, pages 1–8, 2018.

[21] R. Tredinnick, B. Boettcher, S. Smith, S. Solovy, and K. Ponto. Uni-cave: A unity3d plugin for non-head mounted vr display systems. In *2017 IEEE Virtual Reality (VR)*, pages 393–394, March 2017.

Accurate control law for low-cost UAV

Aurélien Cabarbaye,* Titouan Verdu, Jacson Miguel Olszanecki Barth and Gautier Hattenberger
ENAC, Avenue Edouard Belin, France

ABSTRACT

Present article proposes a method to design a control law and the associated observer to stabilize a UAV. The design is only based on the UAV geometry, mass and propulsion system characteristics which do not require any expensive facilities or software to be obtained. The control inputs are the longitudinal airspeed, the roll angle and the slope angle to ease the guidance control whether manual or automatic. The resulting control only relies on the information provided by a 6 Degrees of Freedom (DoF) Inertial measurement unit (IMU) that makes it suitable for implementation on very basic autopilot board (cf. Paparazzi Chimera, Pixhawk XRacer, ArduPilot APM...). The propeller acts indeed as an airspeed probe which makes additional sensor unnecessary. This low-cost implementation makes it of particular interest for large UAVs fleet.

1 INTRODUCTION

The ENAC UAV team has developed, since 2003, the Paparazzi UAV (Unmanned Aerial Vehicle) open-source drone project. It enables to convert quickly and easily an electric powered fixed-wing airframe in a semi-automatic drone. However, just like alternative autopilot, the default control law consists of various PID which gains must be tweaked manually during the first flight tests [1] [2]. Such method turns out to present some issues:

- It is time-consuming
- It requires good piloting capabilities
- It is hazardous since it does not offer any protection against unstable modes
- It requires some practices to identify which gain must be modified and how much it may be tweaked.
- the thrust control does not consider the complex behaviour of the propulsion system.
- The resulting gain adjustment does not usually offer the shortest response time or oscillation attenuation.

*Email address: contact_aurelien.cabarbaye@enac.fr

To solve those issues, the present article proposes a more accurate control. This latter is designed from the aeroplane model detailed in section 2, page 1 whose procurement only relies on free data and calculation tools. The control law is exposed in section 3, page 4. This control depends on the state vector which is estimated by the observer, presented in section 4, page 5. The observer is based on the already known aeroplane model of section 2. It takes thus into account the aeroplane dynamics which makes it more precise than other methods that only rely on IMU data (e.g. complementary, Kalman filters [3]). However, as these alternatives, it is designed to only require information from a low-grade IMU to be implementable on a wide range of low cost existing autopilot boards (e.g. Paparazzi Apogee [4], Pixhawk XRacer [5]) which is of great interest to make up a drone swarm [6]. The resulting control law and observer are tested by simulation in section 5, page 6

The whole process is implemented on the airplane shown on figure 1

Figure 1: Experiment airplane

2 AIRCRAFT MODEL

The model of the UAV is constructed from its dynamics [7]:

$$
\begin{aligned}
&F_X - mg\sin\theta = m\left(\dot{u}^E + qw^E - rv^E\right)\\
&F_Y + mg\cos\theta\sin\phi = m\left(\dot{v}^E + ru^E - pw^E\right)\\
&F_Z + mg\cos\theta\cos\phi = m\left(\dot{w}^E + pv^E - qu^E\right)\\
&Q_L = I_x\dot{p} - I_{zx}\dot{r} + qr\left(I_z - I_y\right) - I_{zx}pq\\
&Q_M = I_y\dot{q} + rp\left(I_x - I_z\right) + I_{zx}\left(p^2 - r^2\right)\\
&Q_N = I_z\dot{r} - I_{zx}\dot{p} + pq\left(I_y - I_x\right) + I_{zx}qr\\
&p = \dot{\phi} - \dot{\psi}\sin\theta\\
&q = \dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi\\
&r = \dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi\\
&\dot{\phi} = p + (q\sin\phi + r\cos\phi)\tan\theta\\
&\dot{\theta} = q\cos\phi - r\sin\phi\\
&\dot{\psi} = (q\sin\phi + r\cos\phi)\sec\theta\\
&\dot{x}^E = u^E\cos\theta\cos\psi + v^E\left(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi\right)\\
&\quad + w^E\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right)\\
&\dot{y}^E = u^E\cos\theta\sin\psi + v^E\left(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi\right)\\
&\quad + w^E\left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right)\\
&\dot{z}^E = -u^E\sin\theta + v^E\sin\phi\cos\theta + w^E\cos\phi\cos\theta\\
&u^E = u + W_x\\
&v^E = v + W_y\\
&w^E = w + W_z
\end{aligned}
\tag{1}
$$

where F_X , F_Y and F_Z are the forces, u, v and w are the velocities, p (roll), q (pitch)and r (yaw) are the rotation speed respectively and W_x, W_y and W_z are the wind speed on the airplane axis X (upstream), Y (right wing direction), Z (downward), m and $I = \begin{pmatrix} I_x & I_{xy} & I_{xz} \\ I_{xy} & I_y & I_{yz} \\ I_{xz} & I_{yz} & I_z \end{pmatrix}$ are the airplane mass and moments of inertia. g is the gravity acceleration. x, y, and z are the earth reference frame axis. ϕ (roll), θ (pitch) and ψ (yaw) are the euler angles between the earth and the airplane reference frames.

The model can be simplified with the following assumptions:

Assumption 1 *Thanks to the relative symmetries around the airplane axis,* $I = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix}$

Assumption 2 *The gusts are supposed to be slow so the wind parameters W_x, W_y and W_z are constant.*

Assumption 3 *Roll and slope angles are likely to be small, in the range :*$(\varphi, \phi) \in [-0.5; 0.5]^2$

Assumption 4 *Roll, pitch and yaw rate angles are likely to be small*

However, the yaw angle (ψ) is not bounded. Therefore, stabilization and guidance must be addressed separately as it is currently done in paparazzi [2].

To solve the issues exposed in section 1, only the stabilization must be modified. Therefore, the state vector is reduced to:

$$
X = \begin{pmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \varphi \\ \theta \end{pmatrix} \tag{2}
$$

Assumption 5 *$v << 1$ $w << u$ which imply that the sideslip angle and the angle of attack are small: $\beta = \tan^{-1}\left(\frac{v}{u}\right) << 1$ (symmetric flight) and $\alpha = \tan^{-1}\left(\frac{w}{u}\right) << 1$ (unstall flight). Therefore, the aerodynamics should present an almost linear behaviour.*

Thanks to assumptions 3 4 and 5, the differences between a linearized model and the real dynamics should be small. The stabilization of the UAV around a flight configuration is therefore based on its linearized model. This latter is obtained in two steps. The main body is first studied. Then the propulsion system is modelled.

2.1 Main body

Athena Vortex Lattice (AVL) is an open source aerodynamics simulation software based on the Vortex Lattice Methods [8]. It assumes an inviscid flow which does not allow good zero-lift drag estimation. However, it seems very accurate to assess aerodynamics evolutions [9]. It is therefore used to generate the linear model of the UAV glider base.

AVL requires information about the shape and the moment of inertia of the aeroplane.

The aeroplane shape is defined with its main aerodynamics characteristic dimensions as shown on figure 2 and with the airfoil shapes.

If the airfoil shapes are unknown, they can be reverse engineered with a profile gauge with sufficient accuracy.

Thanks to assumption 2, the moment of inertia can be easily measured following the Bifilar Pendulum methodology [10] as shown on figure 3

The model provided by AVL is of the form:

$$
\dot{X} = AX + BU \tag{3}
$$

Since the experiment airplane only have two aerodynamic controls (elevons), the control input is defined as:

$$
U = \begin{pmatrix} \delta_{elevator} \\ \delta_{aileron} \end{pmatrix} \tag{4}
$$

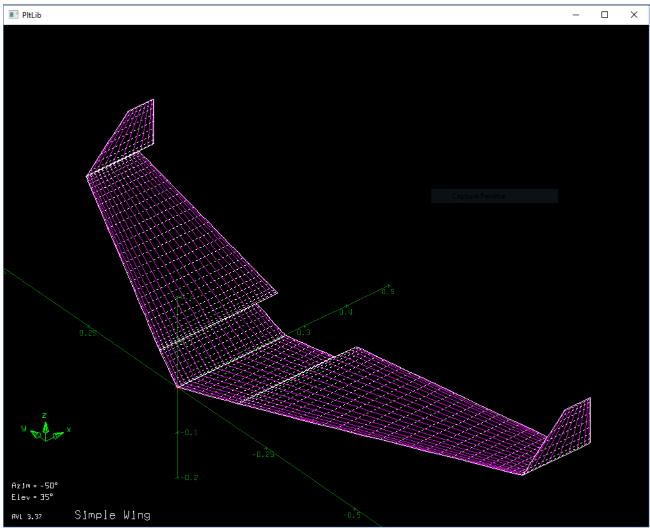


Figure 2: experiment airplane geometry in AVL

Figure 3: Moment of inertia estimation experiment

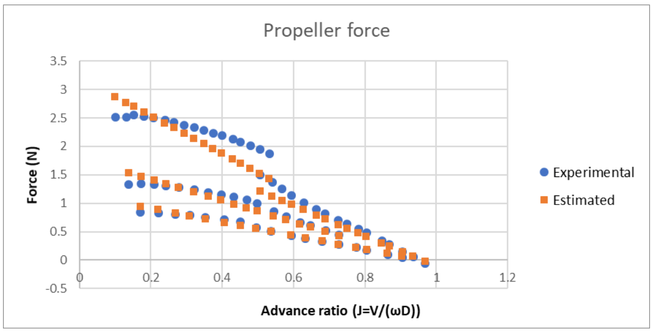


Figure 4: Force function of advance ratio J (where V, ω et D are respectively the airspeed, the rotation speed and the diameter of the propeller)

It must be noticed that the control surface deflections are managed in degree by AVL rather than in radian for the attitude angles and rates. This particularity is kept unchanged in the following of the article. The resulting elevons deflection $\delta_{left\ elevon}$ and $\delta_{right\ elevon}$ is computed as follows:

$$\delta_{left\ elevon} = \delta_{elevator} - \delta_{aileron} \tag{5}$$

$$\delta_{right\ elevon} = \delta_{elevator} + \delta_{aileron} \tag{6}$$

2.2 *Propulsion system*

The propulsion system is composed of a propeller and an electric motor.

Propeller Propeller behaviour can be described by the force F and torque Q produced. F and Q can be very well assessed from the theory resulting in the mixt between Momentum theory and blade element theory [11]. Thanks to assumption 5, the impact of the radial airspeed is neglected. The equations can be thus simplified as follow:

$$F = Kf_\omega \omega^2 + Kf_\chi \omega U \tag{7}$$

$$Q = Kq_\omega \omega^2 + Kq_u U^2 + Kq_\chi \omega U \tag{8}$$

where ω is the propeller rotation speed and Kf_ω, Kf_χ, Kq_ω, Kq_u, $Kq\chi$ are constants that must be estimated from wind tunnel test. Library of such test results are available for a wide range of propellers [12].

The simplified modelled is adjusted to the test results obtained for the APC 8x6E mounted on the experiments aeroplane. The force estimation, as well as the test data, are shown on figure 4. The model estimates very well the experimental data except for those performed in hover (on the top left-hand side). This is because the induced air velocity generated by the propeller is no more negligible compared to the general airspeed. However, this situation is very unlikely to happen in normal flight, so it does not call into question equation (7). The torque estimation and its corresponding test data are shown on figure 5. The very little difference between experimental and estimated data validates equation (8). The propeller proposed model can be ultimately validated with the propeller efficiency: $\eta = \frac{F \cdot u}{Q \cdot \omega}$ shown on figure 6. The results confirm once more the suitability of equations (7) and (8).

Since these equations are very similar, propeller generated force and torque are shown together on figure 7. It can be seen that torque varies in a linear fashion with the force which gives the approximation:

$$\Delta Q = a_{Q/F} \Delta F \tag{9}$$

where $a_{Q/F}$ is a constant.

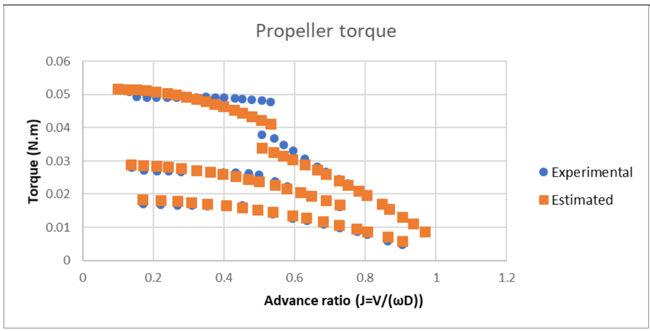


Figure 5: Torque function of advance ratio J

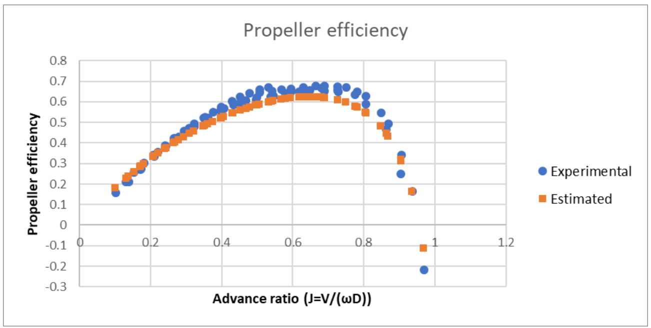


Figure 6: Efficiency function of advance ratio J

Electric motor The electric motor behaviour is assessed thanks to two equations which gives the relation between the electrical and the mechanical parameters [13]:

$$\omega = K_V (V - RI) \tag{10}$$

$$I = K_V Q + I_0 \tag{11}$$

where V and I are respectively the voltage and the current applied to the motor and K_V, I_0 and R are respectively the motor velocity constant, zero load current and electric resistance. The model can be complexified to assess even better the motor behaviour but the adopted one seems to be sufficient [14]. Motor constants K_V, I_0 and R can be easily retrieved if not provided by the manufacturer [15].

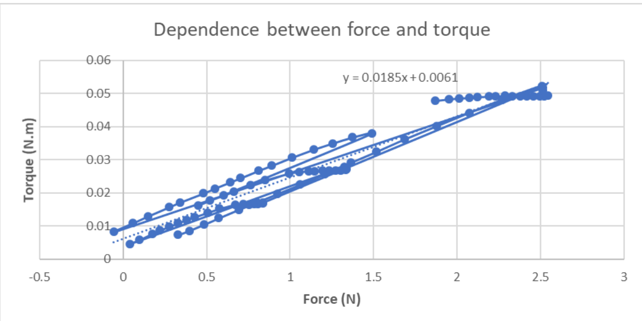


Figure 7: torque function of force

3 CONTROL

The aim of the control is to stabilise the aeroplane to make it as easy to steer as possible whether in manual or automatic guidance mode (cf. $AUTO1$, $AUTO2$ [2]). To do so, the control must integrate a stall protection, fly the aeroplane symmetric and be easily integrated into existing autopilot.

3.1 Control idea

The control states are designed first. Usually, the thrust or the power are directly controlled by the pilot in closed loop fashion. I order to ease the control, a closed loop control is built to stabilise the longitudinal velocity u.

Most aeroplane lateral controls are built to set the roll angle φ to a reference value. This enables an effective path following since there is a direct relation between φ and the turning radius. φ is therefore chosen as the second control state.

For the longitudinal control, there is no consensus. Aircraft are usually controlled in pitch rate, in vertical acceleration or in pitch angle [16]. A control on a derivative (cf. pitch rate, vertical acceleration) imposes more work to the guidance control. A control on pitch angle is not very precise because of the angle of attack variations during the flight. Therefore, the climb slope γ is chosen as third control state. Thanks to assumption 5, γ follows:

$$\gamma = \theta - \alpha \approx \theta - \frac{w}{u} \tag{12}$$

Applying the small disturbance linearization method [7]:

$$\Delta\gamma = \Delta\theta - \frac{\Delta w}{u_0} + \frac{w_0}{u_0^2}\Delta u = \Delta\theta - \frac{1}{u_0}\Delta w + \frac{\alpha_0}{u_0}\Delta u \tag{13}$$

Therefore, the output vector becomes:

$$Y_c = \begin{pmatrix} u \\ \varphi \\ \gamma \end{pmatrix} = C_c X \tag{14}$$

where $C_c = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{\alpha}{u_0} & 0 & -\frac{1}{u_0} & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Three controls input are required to control the three outputs. F is chosen for the third one to removes the non-linearity of equations 7. Therefore, from equation (4) U becomes:

$$U = \begin{pmatrix} \delta_{elevator} \\ \delta_{aileron} \\ F \end{pmatrix} \tag{15}$$

And equation (3) gives

$$\dot{X} = A_c X + B_c U \tag{16}$$

where $A_c = A$ and $B_c = \begin{pmatrix} & \frac{1}{M} \\ & 0 \\ & 0 \\ B & -\frac{a_{Q/F}}{I_{xx}} \\ & 0 \\ & 0 \\ & 0 \\ & 0 \end{pmatrix}$

Notice that the relation between force and torque of equation (7) has been considered to compensate the torque as soon as it is applied, without the delay that would have been otherwise induced by the dynamics.

The controllability matrix $C_m = \begin{bmatrix} B_c & A_cB_c & A_c^2B_c & \dots & A_c^7B_c \end{bmatrix}$ has a rank equal to the length of A_c, so the system is controllable.

3.2 Control law

The control law is based on the method of the steady state tracking which consists in stabilising first all the states of X and then to alter some of them to converge the output to the desired value [17].

The control law is of the form:

$$U = -KX + GR \tag{17}$$

where K and G are two gain matrices ad R is the desired output.

Combining equations (16) and (17) leads to:

$$\dot{X} = (A_c - B_cK)\,X + B_cGR \tag{18}$$

K is first chosen to make all the states stable. Therefore, the poles p_c of the matrix $(A_c - B_cK)$ are set negatives thanks to the pole placement method. In order to offer good stability properties, the poles are chosen from A_c poles. Imaginary parts are kept unchanged, positive real parts are substituted by their opposite and non-sufficiently negative real parts are changed by better ones.

Then the convergence of the output is satisfied as follows:

$$\begin{cases} \lim\limits_{t\to\infty} (A_c - B_cK)\,X(t) + B_cGR = 0 \\ \lim\limits_{t\to\infty} Y_c(t) = \lim\limits_{t\to\infty} C_cX(t) = R \end{cases} \tag{19}$$

The system (19) leads to:

$$G = -\left(C_c\,(A_c - B_cK)^{-1}\,B_c\right)^{-1} \tag{20}$$

3.3 Control saturation and trim

Saturations can be added to consider the servo travel limit and the propulsion maximum power. A trim can also be added to balance the model errors. It is simply implemented modifying the values of the control required to fly in the condition of the linearization. The actual control input U_{trim} is:

$$U_{trim} = \begin{pmatrix} \delta_{elevator} + \delta_{0elevator} \\ \delta_{aileron} + \delta_{0aileron} \\ F + F_0 \end{pmatrix} \tag{21}$$

The deflections of the ailerons are obtained from equations (5) and (6). Resolving quadratic equation (7) to get U_{trim}[3] leads to ω. Then combining equations (8), (10) and (11) leads to:

$$\begin{aligned} V =& \frac{\omega}{K_V} + RK_VKq_\omega\omega^2 + RK_VKq_uU^2 \\ &+ RK_VKq_\chi\omega U + RI_0 \end{aligned} \tag{22}$$

where $U = u + U_0$ and U_0 is the airspeed of the linearization.

Noting that $0 \le V \le V_{batt}$ and $\delta_{\min} \le \delta_{elevon} \le \delta_{\max}$, the control applied becomes:

$$\begin{pmatrix} V \\ \delta_{right\ elevon} \\ \delta_{left\ elevon} \end{pmatrix}_{applied} = \min\left(\begin{pmatrix} V_{batt} \\ \delta_{\max} \\ \delta_{\max} \end{pmatrix}, \max\left(\begin{pmatrix} V \\ \delta_{right\ elevon} \\ \delta_{left\ elevon} \end{pmatrix}, \begin{pmatrix} 0 \\ \delta_{\min} \\ \delta_{\min} \end{pmatrix}\right)\right) \tag{23}$$

3.4 Stall protection

A stall protection can be added to the control input γ. To balance the weight, the lift must be [18]:

$$L\cos(\varphi) = mg\cos(\gamma) \tag{24}$$

which gives

$$\gamma_{max} = \cos^{-1}\left(\frac{\frac{1}{2}\rho U^2 SCl\cos(\varphi)}{mg}\right) \tag{25}$$

where γ_{max} is the limit of the desired output γ when $\frac{1}{2}\rho U^2 SCl\cos(\varphi) \le mg$, verifying $\gamma_{max} \in \left[-\frac{\pi}{2}, 0\right]$. $\frac{1}{2}\rho U^2$ is the dynamic pressure, S is the wing surface and Cl is the lift coefficient. Cl can be for example fixed to fly, as for general aviation, at least at $1.3 \cdot V_s$ in normal flight and at least at $1.1 \cdot V_s$ in final approach (V_s is the stall speed). That is to say: $Cl = \frac{Cl_{\max}}{1.14}$ in normal flight and $Cl = \frac{Cl_{\max}}{1.05}$ in final approach.

4 Observer

A good estimation of the state vector X is required by the control law (cf. equation (17)). An observer is therefore built to provide it.

4.1 Baseline idea

To be as low cost as possible to implement, the observer only relies on the information provided by a 6 DoF IMU. The output vector Y_o of the observer is therefore:

$$Y_o = \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ p \\ q \\ r \end{pmatrix} = C_oX + D_oU \tag{26}$$

where $C_o = \begin{pmatrix} \multicolumn{8}{c}{A(1:3,1:8)} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ and $D_o = \begin{pmatrix} \multicolumn{3}{c}{B_0(1::3,1::3)} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ The Observability matrix $O_m = \begin{bmatrix} C_o \\ C_oA_c \\ C_oA_c^2 \\ \dots \\ C_oA_c^7 \end{bmatrix}$ has a rank equal to the length of A_c, so the system is observable.

The dynamics of the simulated observer is [19]:

$$\dot{\hat{X}} = A_c\hat{X} + B_cU + L\left(Y_o - \hat{Y}_o\right) \tag{27}$$

where $\hat{Y}_o = C_o\hat{X} + D_oU$ From equations (16), (26) and (27), the estimation error $\tilde{X} = X - \hat{X}$ dynamics is:

$$\dot{\tilde{X}} = \dot{X} - \dot{\hat{X}} = (A_c - LC_o)\,\tilde{X} \tag{28}$$

To urge the $\tilde{X}$ to zero, matrix L is chosen following the pole placement method with all the poles p_o of $(A_c - LC_o)$ negatives.

4.2 *Observer improvements*

As it will be shown in section 5, page 6, the approximation introduced in equation (7) produces an error of estimation. To improve the observer, the non-linearity of the propulsion system is considered. Equation (27) is thus modified as follows:

$$\dot{\hat{X}} = A_c\hat{X} + B_oU + EQ + L\left(Y_o - \hat{Y}_o\right) \tag{29}$$

where $B_o = \begin{pmatrix} B & \begin{matrix} \frac{1}{M} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} \end{pmatrix}$ and $E = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{-1}{I_{xx}} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ and matrix L obtained in section 4.1, page 5 is kept unchanged. Q and F are computed thanks to equations (7) and (8) where ω is computed solving quadratic equation (22), where V is the result of equation (23).

5 Simulation

A simulation is performed for the experiments aeroplane. The linearization has been made for $u = 10\ m \cdot s^{-1}$ in trimmed steady flight. The resulting data considered for the simulation are exposed in Appendix A:, page 8.

5.1 *Control results*

The control input of equation (17) is fixed to:

$$R = \begin{pmatrix} 5 \\ -0.5 \\ 0.5 \end{pmatrix} \tag{30}$$

The initial state is fixed arbitrarily to:

$$X_0 = \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0.5 \\ 0 \end{pmatrix} \tag{31}$$

States evolution is shown on figure 8. All the states are stable

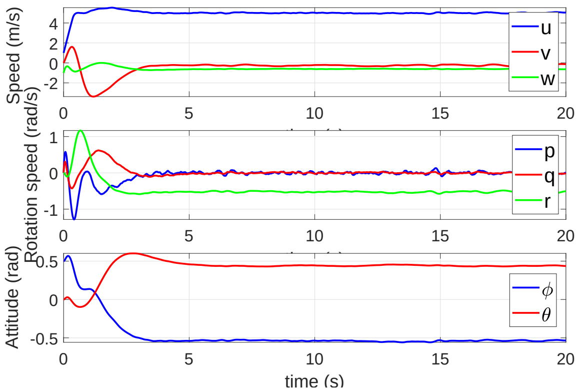


Figure 8: States X

and converge asymptotically which confirms that the control law (17) can perform its first task. The control output Y_c, defined in equation (14), is shown on figure 9. It converges well

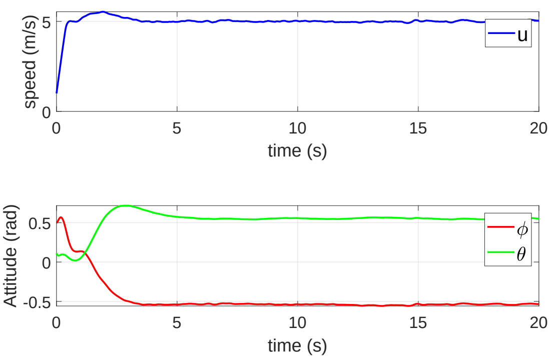


Figure 9: Control output Y_c

to R fixed in equation (30), which confirms that the control law (17) performs its second task.

The control U is shown on figure 10 The saturations work

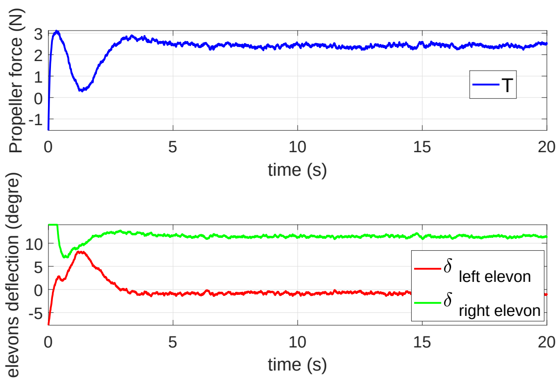


Figure 10: Control U

properly (cf. $\delta_{right \cdot elevon}$).

5.2 *Observer results*

The observer is based on the information given by an IMU. This latter is supposed to be very noisy. To simulate it properly, a white noise has been added to each state of equation (26). The resulting collected information is shown on figure 11 The integration of the observer starts with the estimate states $\hat{X}$ arbitrarily set to:

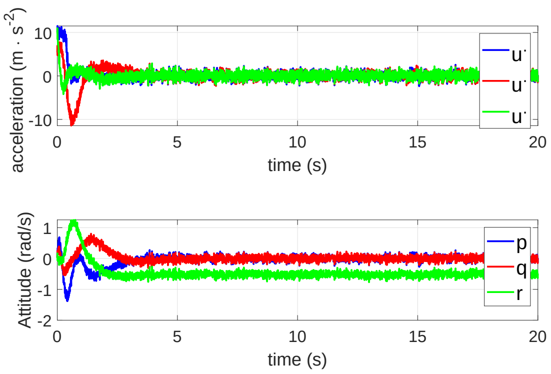


Figure 11: Observer output Y_o provided by IMU

$$\hat{X}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{32}$$

Figure 12 shows what would have been the estimation error of the observer if it has been kept in its baseline form defined in equation (27). It can be noticed that if it seems to

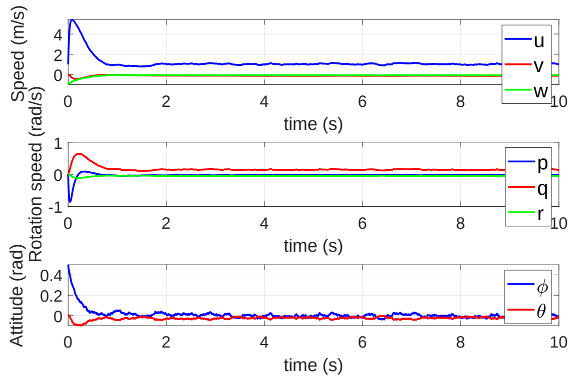


Figure 12: Estimation error $\tilde{X}$

converge, a bias remains, in particular for the airplane longitudinal airspeed u. That is why the observer is improved in section 4.2, page 6. Figure 13 shows the estimation error of the final observer defined by equation (29). All the states

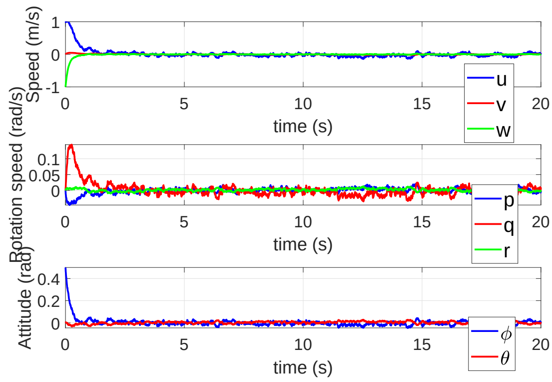


Figure 13: Estimation error $\tilde{X}$

of equation (2) are precisely estimated. u estimation is very little bias even though no specific sensor has been used (e.g. pitot tube). That can be explained by the propeller which depends a lot on airspeed u (cf. equation (7)) and therefore acts as an alternative airspeed probe.

Moreover, the estimation errors converge very rapidly to zero (less than a second) which makes it able to compensate for most of wind gusts. If a faster convergence is needed, the poles p_o can be increased.

6 Conclusion

The aircraft model has been built in section 2 in two steps. First, the linearized main aeroplane body model is obtained from AVL software. Then the non-linear propulsion system model is adjusted from freely available wind tunnel tests. A simplification of the propeller model enables to build a linear overall model that is used as a starting point for the control law design in section 3. This control law focusses on airspeed u and angles ϕ and γ to ease the guidance whether manual or automatic. The states knowledge required by the control is provided by an observer designed in section 4. This observer is based on the non-linear model resulting from the mix between the linear main body model and the non-linear propulsion model, rather than on the linear model used for the control law design. This enables to track much better the states and in particular, the longitudinal speed since the propeller plays the part of an airspeed probe. In addition, the observer high convergence speed makes it suitable to deal with wind gusts. Tests of control law and observer are performed by simulation in section 5. Results seem to prove the suitability of the proposed method.

References

[1] PX4 Dev Team. Fixedwing pid tuning guide, August 2018.

[2] ENAC Drone Team. Subsystem/control, December 2016.

[3] Jossué Cariño Escobar, Aurélien Cabarbaye, Moisés Bonilla Estrada, and Rogelio Lozano. Quaternion kalman filter for inertial measurement units. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, pages 1037–1043. IEEE, 2017.

[4] ENAC Drone Team. Apogee/v1.00, August 2018.

[5] PX4 Dev Team. Pixracer, September 2018.

[6] Andrew W Sanders. Drone swarms. Technical report, US Army School for Advanced Military Studies Fort Leavenworth United States, 2017.

[7] B. Etkin. *Dynamics of Atmospheric Flight*. Dover Books on Aeronautical Engineering. Dover Publications, 2012.

[8] Mark Drela. Avl 3.36 user primer, February 2017.

[9] Ramon Lopez Pereira. Validation of software for the calculation ofaerodynamic coefficients: with a focus on the software package tornado. Master's thesis, Linkpings Universitet, 2010.

[10] Matthew Jardin and Eric R. Mueller. Optimized measurements of uav mass moment of inertia with a bifilar pendulum. 46:763–775, 05 2009.

[11] W. Johnson. *Helicopter Theory*. Dover Books on Aeronautical Engineering. Dover Publications, 2012.

[12] J.B. Brandt, G.K. Deters, R.W. andAnanda, and M.S. Selig. *UIUC Propeller Database*. University of Illinois at Urbana-Champaign, August 2018.

[13] Mark Drela. *First-Order DC Electric Motor Model*. MIT Aero & Astro, February 2007.

[14] Mark Drela. *Second-Order DC Electric Motor Model*. MIT Aero & Astro, March 2006.

[15] Mark Drela. *Measurement of Brushed DC Electric Motor Constants*. MIT Aero & Astro.

[16] P.G. Hamel. *In-Flight Simulators and Fly-by-Wire/Light Demonstrators: A Historical Account of International Aeronautical Research*. Springer International Publishing, 2017.

[17] R.L. Williams and D.A. Lawrence. *Linear State-Space Control Systems*. John Wiley & Sons, 2007.

[18] D.P. Raymer. Aircraft design: A conceptual approach, 2006.

[19] C.T. Chen. *Analog and Digital Control System Design: Transfer-Function, State-Space, and Algebraic Methods*. OUP USA, 2006.

Appendix A: Data

$$Kf_\omega = 6.49343E - 06; \tag{33}$$

$$Kf_\xi = -0.000209663; \tag{34}$$

$$Kq_\omega = 1.03864E - 07; \tag{35}$$

$$Kq_\xi = 5.02077E - 07; \tag{36}$$

$$Kq_u = -9.98976E - 05; \tag{37}$$

$$a_{Q/F} = 0.0185 \tag{38}$$

$$K_v = 1200; \tag{39}$$

$$I_0 = 0.1; \tag{40}$$

$$R = 0.165; \tag{41}$$

$$a_{Q/F} = 0.0185 \tag{42}$$

$$A = \begin{pmatrix} -0.2153 & -0.0001 & 1.61 & 0.0003 & -1.0409 & 0 & 0 & -9.81 \\ 0 & -0.5008 & 0.0005 & 1.1056 & 0 & -9.7908 & 9.81 & 0 \\ -0.8621 & 0 & -9.4525 & 0 & 8.2192 & 0 & 0 & 0 \\ 0 & -5.76980 & -13.6429 & 0 & 2.3098 & 0 & 0 & \\ 0.7595 & 0 & -7.3621 & 0 & -4.8625 & 0 & 0 & 0 \\ -0.0003 & 0.702 & -0.0154 & -2.4765 & -0.0011 & -0.309 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{43}$$

$$B = \begin{pmatrix} 2.42E - 02 & 2.74E - 05 \\ 1.33E - 05 & -3.56E - 02 \\ -0.5802 & 7.61E - 09 \\ -6.95E - 07 & -3.064 \\ -3.421 & -5.71E - 07 \\ -3.06E - 04 & -5.95E - 03 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{44}$$

$$p_c = \begin{pmatrix} -7.1559 + 7.4942i \\ -7.1559 - 7.4942i \\ -13.8157 \\ -1.093 + 1.1063i \\ -1.093 - 1.1063i \\ -3.158 + 4.6121i \\ -3.158 - 4.6121i \\ -1 \end{pmatrix} \tag{45}$$

$$K = \begin{pmatrix} -0.8080 & -0.8512 & 1.7957 & -1.8668 & -0.9899 & 0.3489 & -4.6903 & -3.0280 \\ -2.1717 & 1.2099 & 0.6040 & 0.3691 & 0.0817 & 2.9449 & -10.2553 & 1.7359 \\ 7.4471 & -0.0382 & -4.6566 & 0.0428 & -0.2710 & 0.4464 & -2.1005 & -14.0328 \end{pmatrix} \tag{46}$$

$$G = \begin{pmatrix} -0.4752 & -4.7156 & -3.0280 \\ -2.3042 & -6.9212 & 0.6674 \\ 8.5137 & -1.6796 & -4.2228 \end{pmatrix} \tag{47}$$

$$p_o = 2 \begin{pmatrix} -1 \\ -1.5 \\ -2 \\ -2.5 \\ -3 \\ -3.5 \\ -4 \\ -4.5 \end{pmatrix} \tag{48}$$

$$L = \begin{pmatrix} 1.3943 & -0.0794 & -0.0633 & 2.5619 & 17.2684 & -1.0672 \\ -0.3175 & 0.9591 & -0.1471 & -5.7566 & -0.4992 & 0.3051 \\ -0.0334 & 0.0123 & 0.0628 & -0.3708 & 6.3127 & 0.1625 \\ 0.2940 & 0.0375 & 0.1486 & -1.7129 & 0.3492 & 1.4643 \\ 0.0993 & -0.0269 & 0.7555 & 0.6391 & -0.5620 & -0.3317 \\ -0.0357 & -0.0046 & -0.0165 & -3.6842 & -0.0503 & 1.7936 \\ 0.0332 & 0.5900 & -0.0037 & 0.0501 & -0.0625 & 5.8123 \\ -0.4986 & -0.0451 & -0.0852 & -0.2193 & 0.4353 & -0.4092 \end{pmatrix} \tag{49}$$

ARChEaN: Aerodynamics of Rotors in Confined ENvironments
Study in Ground and Corner Effect

S. Prothin, C. Fernandez Escudero, T. Jardin, and N. Doué
ISAE-Supaero, Université de Toulouse, France

ABSTRACT

The work presented in this paper is part of a project called ARChEaN (Aerodynamic of Rotors in Confined ENvironment) whose objective is to study the interactions of a micro-drone rotor with its surroundings in the case of flight in enclosed environments such as those encountered, for example, in archaeological exploration of caves. To do so the influence of the environment (walls, ground, ceiling…) on the rotor's aerodynamic performance as well as on the flow field between the rotor and the surroundings will be studied. This paper will focus in two different configurations: flight near the ground and flight near a corner (wall and ground) and the results will be analysed and compared to a general case consisting of flight far away from any obstacle. In order to carry out this analysis both a numerical and an experimental approach will be carried out in parallel. The objective is to validate the numerical model with the results obtained experimentally and to benefit from the advantages of both approaches. This research work is an important step as it will lead to knowledge on how to operate these systems as to minimise the possible negative environment disturbances, reduce fuel consumption and predict the micro drone's behaviour during its enclosed flights.

1 INTRODUCTION

Since their first developments, drones (unpiloted aircrafts) have revolutionised flight as we know it, opening a wide range of new possibilities which were unconceivable some decades ago. Drones allow us to go further than ever and their applications, which go from military uses to observation, exploration, meteorology, audio-visuals etc., are nowadays growing exponentially in parallel to new technological developments. Their versatility and flexibility make them a great sector in which to carry out new research and development projects.

The study focuses on how a micro rotor of a drone at stationary flight interacts with its environment. This will lead to improving the efficiency of this type of drone and to reducing the effects of detrimental phenomena on the local environment. The aerodynamic forces experienced by the micro rotor will be measured and the velocity and pressure distribution in the fluid surrounding it will be evaluated. The analysis and comprehension of this phenomenon will allow us to create models that represent the physics involved in these situations to integrate them in the drone's control laws.

The idea of this paper is to carry out new configuration cases and to study them using both a numerical and an experimental approach in parallel in order to compare the two of them and to benefit from the advantages that each one has to offer.

*contact email: sebastien.prothin@isae.fr

In this document, the context of the study will be explained, the two study approaches presented in order to explain and analyse the results obtained with each one and, finally the two will be compared. A list of nomenclature and a list the figures presented as well as the bibliography studied is included at the end in order to aid the reader in his/her understanding of this text.

In this project it was found that very few previous investigations had been carried out regarding the aerodynamics of micro-drone flight in closed environments which means, on one hand, that any results found will be of great interest as it is a very new study field with many applications but, at the same time that not much guidance is available. Regarding the out of ground effect (OGE) and the in ground effect (IGE), the first study cases of this study; there have been a significant number of studies carried out. However, the vast majority consider helicopter flight which means that the results are valid for large Reynolds number and since this project involves drones, small Reynolds numbers and since this project involves drones, small Reynolds numbers are involved so the results of these investigations do not necessarily apply. Furthermore, no studies have been found regarding flight in corner effect.

Although the complete list of bibliographical references consulted during the development of this project are presented at the end of this paper.

2 GEOMETRY AND CONFIGURATIONS

2.1 Rotor geometry

As previously explained, two approaches will be carried out in the present project. An experimental approach will be made in order to encounter the real phenomena and, additionally, CDF simulations will be developed to achieve a more thorough knowledge of the aerodynamics involved. The two approaches will be constantly compared and validated with one another.

The geometry of rotor selected for every test carried out in this project is a two-blade rotor with rectangular blades of 100mm in length and 25mm in chord (c), having an aspect ratio (AR) of 4. As the figure 1 shows, the radius of the rotor (R) is 125mm. Similarly, this rotor geometry will be reproduced in the numerical simulations. The blades have a constant angle of attack of 15 degrees.

2.2 Confined configurations

A closed environment is complicated to simulate as it includes many variables regarding the types of obstacles encountered, the distances between the obstacles from each other and from the drone etc. In order to simplify the problem and be able to study the phenomena in a general way, different study cases have been chosen to simulate different enclosed configurations. Since the ARChEaN project started in 2014 various configurations have been tested before arriving to the configurations analyzed throughout this report. In figure 2, all these configurations will be briefly listed and explained.

- **Off ground effect (OGE)**: the reference case without obstacles.
- **In Ground Effect (IGE)**: presence of a wall downstream of the rotor.
- **In Ceiling Effect (ICE)**: presence of a wall upstream of the rotor.
- **In Wall Effect (IWE)**: presence of a wall perpendicular to the rotor plane.
- **In Channel Effect (IChE)**: presence of two walls, one upstream and the other downstream of the rotor.
- **In Low Corner Effect (ILoCE)**: presence of two walls, one downstream of the rotor and the other perpendicular to the rotor plane.
- **In Upper Corner Effect (IUpCE)**: presence of two walls, one upstream of the rotor and the other perpendicular to the rotor plane.

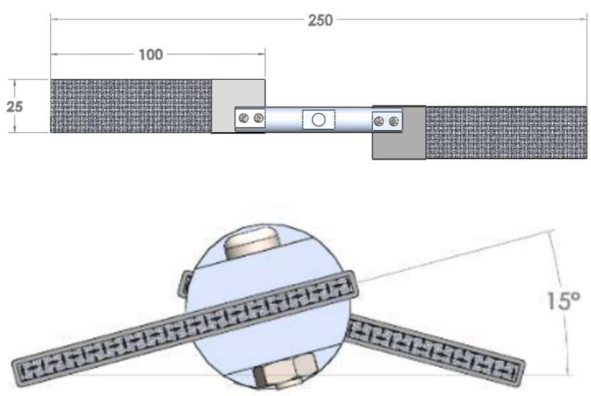


Figure 1 – Rotor geometry

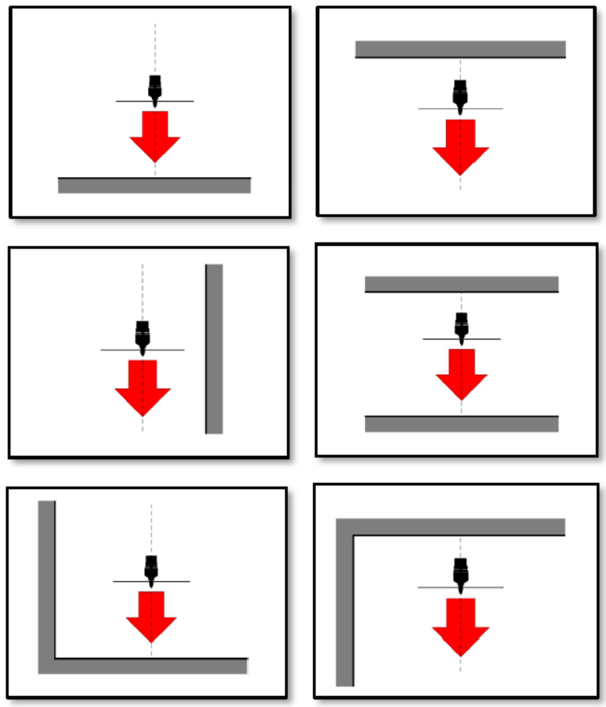


Figure 2 – Wall's configurations

In this paper we will present only ILoCE, cases (OGE, IGE, ICE, IChE) were studied and presented in **[16]**.

The configuration studied here is the in low corner effect (ILoCE) which, similarly, refers to the particularities encountered by an aircraft when it is flying or hovering close to the ground or over a flat surface and close to a perpendicular wall or flat surface simultaneously. Again, every test and simulation was conducted in hovering flight conditions.

Within this configuration, three different cases will be analysed each with a different rotor to ground and rotor to wall distance. To characterise each case the dimensionless magnitude h/R, previously explained in the IGE configuration will be used, and we will introduce a new dimensionless magnitude, d/R where d in the distance from the centre of the rotor to the wall and R, again, is the rotor length.

The cases studied within this configuration will therefore be referred to as:

- ILoCE h/R=2 d/R=2
- ILoCE h/R=2 d/R=3
- ILoCE h/R=3 d/R=2

Similarly to in the IGE configuration, the objective is to fly at a constant lift value of 2N. Therefore, experimental tests on each of these configurations will also be carried out to obtain the rotation speeds required which will then be imposed in the numerical simulations, validated and used to obtain the blade effort distribution and the behaviour of the fluid around the rotor. Additionally, in the ILoCE case, further experimental analysis of the wake will be carried out using PIV (particle image velocimetry) measurements. Further information on both the numerical and experimental approaches will be detailed in the corresponding sections further on.

3 EXPERIMENTAL APPROACH

3.1 Efforts measurements

As mentioned before, all the configurations in this paper have been carried out considering a constant value of the total lift of 2N. The rotation speed corresponding to this lift value changes significantly depending on the configuration studied. These speeds have therefore been obtained experimentally. The apparatus used to obtain the experimental values plays a crucial role in the results obtained. Therefore, the main components will be explained as well as their importance in the experiment.

The rotor frame of reference is shown below to help understand the setup. The balance has a different frame of reference as will be explained further on but the measurements will be converted to rotor frame of reference since the results will be expressed as seen by the rotor, figure 3.

It is the main measurement tool which allows the forces and moments in three directions to be measured. The balance, showed in figure 4, has a cylindrical shape. It is 21.6mm long and has a diameter of 25 mm. This balance is a non-intrusive object since it doesn't cause any perturbation to the flow field due to its perfect integration in the experimental setup. The sensing range of the balance is 125N for the forces along the X and Y axis, 500N along the Z axis and 3Nm for each moment. The maximum error in a measurement given by the manufacturer, and verified in our laboratory, is of 1 % in force and 1.25 % in torque.

A 350W brushless motor, the MikroKopter® MK3638, is connected to the rotor providing it with the necessary power for rotation.

To simulate the presence of the ground we use a table. This table of measurements is composed of nine PMMA detachable plates of 30 by 30cm constituting a 90 by 90 cm surface. Similarly a wall made out of a single piece is placed perpendicularly to it to simulate the wall.

To place the rotor at different distances we use a displacement system. A steel support rod was used to join the displacement system and the motor ensemble. The data of each measurement point was acquired at constant rotational and static conditions reproducing rotor hovering. Four different measurements were done for each set velocity-position to verify the measurement's repeatability. During the tests, the blade's azimuthal position was acquired and the atmospheric conditions (Tatm, Patm) measured.

The convergence of signal-spectrums for every variable is checked. If the correct convergence is showed, the mean of the 50000 samples is calculated for every parameter.

3.2 PIV measurements

As previously mentioned in the geometry and configuration section, a PIV analysis will be carried out to achieve an experimental knowledge of the behaviour of the fluid surrounding the rotor in the ILoCE cases.

For each ILoCE case the rotor plane of symmetry, which is perpendicular to the ground and to the wall, will be analyzed as well as other planes parallel to it to obtain a 3d velocity distribution of the fluid volume centered around the rotor, figure 5.

The tracer particles chosen for this experiment were olive oil particles (mean diameter of 1 micron) produced by a generator TOPAS ATM 210 H. The mixing of the particles has been carried out in the whole closed room where the experiments took place before the acquisition phase to ensure the flow near the rotor was not being affected.

The laser used is these test is the DualPower Bernouilli PIV 200-15. It is a pulse laser with double cavity which emits light at 532 nm with a cadence of up to 2 x 15 Hz. It delivers an energy pulse of 2 x 200mJ which is spread into a layer with the help of a cylindrical lens. For the acquisition of two consecutive double images each cavity emits a pulse with a temporal gap which corresponds to the time between two images dt= 100μs to have a maximum image displacement of 7 to 8 pixels maximum. The thickness of the layer is an important parameter which in our case is 1.5mm.

For each measurement plane, 1000 pairs of stereoscopic PIV images were captured in the first campaign with two high-resolution cameras. Each image of 16MP is captured at a frequency of 2Hz, with a time of 80µs between two consecutive images. The cameras have a Scheimpflug setup which enables them to work in conditions such that the object, the objective plane and the image plane have a common axe. This way the whole plane can be captured without having requiring a bigger lens aperture.

The software "DynamicStudio v4" developed by Dantec Dynamics is used for correlate these groups of four images (two of each camera), in order to find the velocity vectors of these particles and to determine the actual flow field. An overall view of the setup is shown in figure 6.

4 NUMERICAL APPROACH

The numerical calculations carried out in this project are computed by the CFD solver STARCCM+. The objective is to solve the URANS (unstationary RANS) incompressible equations using the gradient method (hybrid gauss-LSQ) which is an implicit unstationary discretisation schema by finite volumes of second order in time and space. The turbulence model used is the Spalart-Allmaras model which will be briefly explained in the following section.

Since the flow is incompressible, a segregated flow approach is used instead of a coupled flow approach which would take a much longer time. Also, since the flow is incompressible and isothermal, constant density applies which also saves time.

4.1 Boundary conditions

The appropriate boundary conditions must be defined in the limits of the domain defined in the calculations.

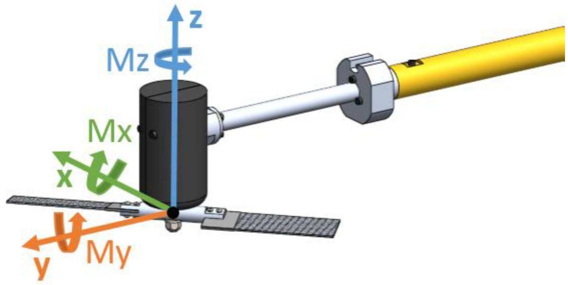


Figure 3 - Rotor frame of reference

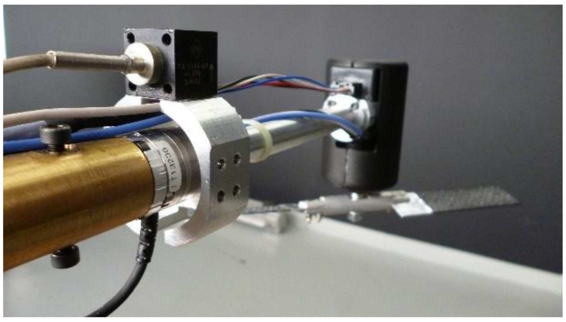


Figure 4 – Six-Component Force/Torque balance (ATI Nano 25)

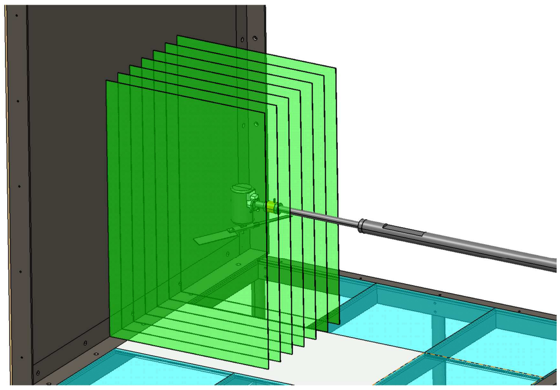


Figure 5 – PIV plane

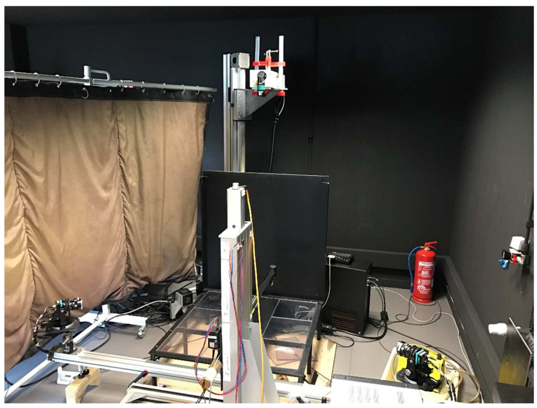


Figure 6 – PIV setup

In our three cases, different combinations of the following different boundary conditions were consequently imposed, table 1:

- Stagnation inlet: far field condition in which the flow is at rest.
- Pressure outlet: defines the static pressure at the exit.
- Wall: represents an impermeable surface which separates the fluid and the solid medium. A slip condition is imposed.

4.2 Meshes

In the present case, the two blade rotor presented in previous sections has been reproduced with the exact same geometry. The two blades were designed using the software Gambit-ANSYS. Only the two blades of the rotor were created, without the hub, in order to actually analyse the aerodynamics of the blades without the influence of any other elements. The mesh around the blades is also created in Gambit. This mesh, called 3D-"O-Grid Mesh", contains cells starting at 45º in any direction from the blade surface with face sizes of 1-2mm² in the first layers of the boundary layer mesh. Each one of these meshes will constitute an individual fluid region which will be rotating together with the blades. This region contains around 920,000 cells and it dimensions are 125mm in length and 25 by 50mm in the chord plane.

The mesh of the fluid volume was created separately in StarCCM+®, a CD Adapco® commercial CFD solver. This mesh is generated using a trimmer meshing technique. This method allows to create high-quality meshes for simple or complex geometries, using hexahedral cells.

Once both volumes have been meshed, the mesh around the blades and then blades are imported into StarCCM+®, figure 7, figure 8.

It is crucial to pay attention to the size ratio between the mesh around the blades and the domain mesh. Where the two meshes are in contact the size of both must be very similar. Therefore, the mesh near the rotating volume will be 1mm and will grow progressively to reach a size of 5 mm within the domain.

Conditions	**OGE**	**IGE**	**ILoCE**
Inlet	Stagnation inlet	Stagnation inlet	Stagnation inlet
Outlet	Pressure outlet	Wall	Wall
Laterals	Pressure outlet	Pressure outlet	3 Pressure outlets + 1 wall

Table 1 – Boundary conditions for different study cases

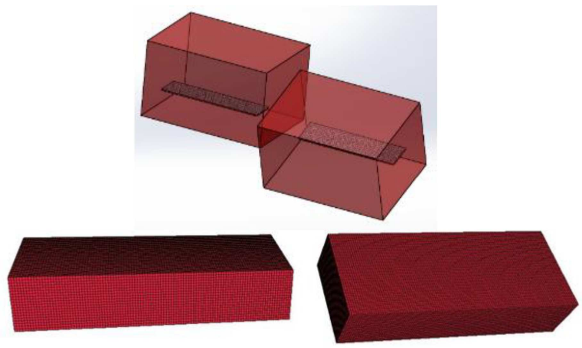


Figure 7 - Blade's Region mesh in CFD simulations

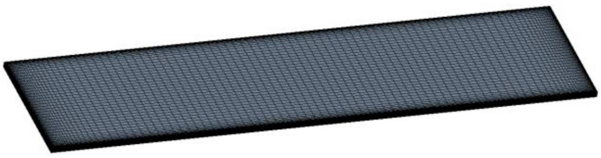


Figure 8 - Blade surface mesh

The trimmer mesh of the external region and the O-grid rotating mesh of the blade's region should be connected to allow the fluid exchange between both regions. This connection is carried out using the "Chimera Grid Method" or "Overlap Method", which is known as Overset in StarCCM+®, which makes it possible to overlap two structured meshes enhancing the local resolution of the meshes. A continuous interpolation from one mesh to the other is done at each time step.

The control volume used for the OGE and IGE cases has a size of 250x500x250 mm. For the OGE case, the rotor is placed at a distance of one radius below the horizontal upper wall which corresponds to the stagnation inlet limit condition and at a distance of four radius of the lower wall which corresponds to the pressure outlet limit condition. In the IGE configurations the lower and upper walls are positioned at the different desired simulation distances.

To ensure the convergence, before our meshes were chosen, calculations were carried out with a finer and a coarser mesh size as well as with a shorter and a longer time step. Every rotor turn is divided into 180 time steps and each time step has 20 iterations. Each simulation has been carried out until 100 rotor turns were reached in the ILoCE cases and 50 rotor turns in the IGE cases. For each case, the convergence of the calculations has been checked by looking at the values of the blade forces reached throughout the simulations. Making a convergence test of this type is crucial in any numerical simulation as a non-converged simulation could lead to erroneous results, figure 9.

5 RESULTS

5.1 Lift Values

As previously explained, to ensure that the results correspond to a constant lift value of 2N, the rotation speed for each case was found experimentally, this value was imposed in the numerical simulations and the results obtained were compared to those obtained experimentally. These results are shown in this sections and the proximity of both approaches can be remarked, table 2 & 3.

It can be observed that the numerical approach tends to underestimate the lift value as most of the experimental lift values are below 2N. Unfortunately, a technical problem regarding the balance described in the experimental section took place right after the RPM measurement for the ILoCE h/R=2 d/R=2 was recorded. Therefore this RPM was imposed in the rest of the ILoCE cases and it was checked in that the lift remained very close to the desired 2N in the numerical simulations. This RPM had a value of 4041.

5.2 Force distribution on blades

IGE case

In this section the lift and drag distributions along the blade is examined through the numerical approach in order to understand its behaviour in relation with the rotor's distance from the ground and/or wall. To obtain a constant lift value of approximately 2N, the values from experimental measurements had been used to find the rotation speed required to achieve the desired lift. This speed is then used in StarCCM+ where it is not possible to enter the lift directly. This means that the area under each mean lift graph is always approximately 2N if both blades are considered. The figure 10 obtained by calculating the mean forces on the last turn after 50 rotor turns had been reached to ensure convergence. The reference case is the OGE and as expected, the closer the rotor is placed to the ground the more observable its influence becomes on the blade force distribution. Similarly, the IGE h/R=2 force distribution is extremely close to that of the OGE case.

Also, while the lift at ¾ blade span from the h/R=1 has a small change from the h/R=0.5, the change near the hub is very noticeable. The h/R=0.25 and the h/R=0.5 cases are far from the rest of the curves at ¾ blade span but similar to one another. Additionally, it is important to observe how the closer the rotor is to the ground the smaller the mean lift on the ¾ blade span is. The opposite is true as we approach the centre of the rotor. Therefore, it can be affirmed that the closer the rotor is to the ground the more uniform the mean lift distribution becomes. The same observations are true for the mean drag distributions.

ILoCE case

Figure 11 clarifies some aspects of the results that will be presented in this section. Firstly, that a rotation begins with the rotor perpendicular to the wall and that we call blade 1 the blade that is closest to the wall in this position. Secondly, that the rotation is anticlockwise and the angle of rotation is measured as shown in the diagram above. In the case of the ILoCE case figures 12 show that the total lift and drag values of blade 1 decrease as the blade moves away from its initial position perpendicular to the wall until it reaches a position parallel to the wall where the force increases again. This means of course, that simultaneously, the forces on blade 2 increase because a total 2N global lift force is achieved. It can be observed that the effect is most noticeable at ¾ of the blade since the forces at the tip and near the hub remain practically constant. The corner effect can also be represented as shown in figure 13 which shows the force distribution as seen from above for different rotor radius. The nonsymmetrical effect caused by the presence of the wall can be observed only for the larger radius numbers. Similarly, this effect can be observed in the force colour maps in figures 14 were it can be observed that the dissymmetry on each blade is the opposite than in the other one.

5.3 Wake

Although the analysis of the fluid flow doesn't lead to a direct quantification of the effects seen by the rotor it results in understanding of the physical phenomena that occurs and it helps us visualise the fluid-structure interaction directly. In this section, diagrams of iso-velocity in a plane section will be shown with the third velocity component represented with colour as to obtain a three-dimensional representation of the flow. The topology of the flows will be commented and compared and in the case of ILoCE the PIV results will also be presented.

OGE case

The reference case, the OGE, figure 15 shows us a perfectly symmetrical flow as expected. The transverse velocity shows us the anticlockwise rotation of the rotor with fluid coming in the plane at the right and out the plane at the left.

IGE case

The four IGE cases show how the fluid wake is deformed by the presence of the ground. For example for case h/R=2, figure 16. The effect is less visible as the rotor moves away from the ground and the IGE h/R 2 case shows that the effect of the ground right under the rotor is almost inexistent. However the other cases show how the jet is squashed due to the presence of the ground and how the flow recirculates upward and meets the rotor from underneath. It is important to remember the fact that the hub has not been simulated which means that some figures show the flow passing upwards in between the blades which is something that would not occur in a real rotor with a blade hub.

ILCoE case

In figure 17 left, we can observe the topology of the fluid wake in corner effect. A toroid can be observed in the blade hub area because the fluid goes into the plane at the right of the hub and out of the plane in the left. Clearly in this case the flow is not symmetrical. Two vortices can be observed underneath the rotor and a source or drain point is formed due to the wall. The interaction with the wall also causes this vortex to rise above the rotor and to interact with it form above. Similarly, figure 17 right shows the same figure with the results obtained from PIV. It can be observed that the PIV captured the same drain/source near the wall and the way that the vortex rises above the rotor. However, the vortices near the ground do not appear. This is because with 1000 PIV acquisitions convergence of velocity derivatives is reached in almost all the region but it not reached near the masked part and the wall and ground. Therefore, a second PIV has been started with the objective of reaching 2500 acquisitions and allowing a more accurate study of the flow.

Another useful way to analyse the flow and the effect of the walls is to analyse the velocity profils. The effect of the wall and ground is particularly noticeable as we approach the corner. The boundary layer of the ground and of the wall can also be observed and it resembles a flat jet boundary layer so the it could be interesting to further investigate the similarities between both.

6 CONCLUSION AND FUTURE WORK

Due to their small dimensions and versatility, drones are can be designed to perform missions in confined environments. Many sectors, such as archeology and nuclear security could greatly benefit from these developments. A way to enhance flight robustness is to understand how the rotor of a hovering drone responds to wall proximity in terms of its effect on the flow field, the force distribution it perceived and its aerodynamics performances. These three effects have been studied in this work and the physical link between them has begun to be analyzed to improve the physical understanding of confined flight.

It is important to understand that the work presented in this report is part of a bigger project were a lot of work has already been done and that there are still many options to investigate further. However, with this work we are one step closer to the final objective of designing and controling drones flying in enclosed envorionments.

As it has been mentioned in the results rection, another PIV campaign is being carried out to reach more accurate experimental results. Also new numerical simulations have already been launched for other corner configurations. The idea is to study the corner effect in us much detail as the ground effect.

Additionally, the other configurations mentioned in the geometry and configurations section such as the wall efect and the channel effect will also be researched to understand as many configurations as posible to be able to eventually fly drones in real enclosed environments. These configurations will continue to be analysed both numerically and experimentally. The PIV configuration will probably be changed to allow a more complete analysis of the fluid plane and the balance which had a problem will be recalibrated or replaced.

Another possible future step would be to move on from the static flight analysis and analyse the effect of different obstacles on a rotor carrying out dynamic flight. This could be done experimentally by replacing the displacement rod described in the experimental setup section by a robotic arm which would move the drone as commanded.

REFERENCES

[1] A. Betz. The ground effect on lifting propellers. NACA TM 836, NACA.

[2] Knight M. et Hefner R.A. Static thrust analysis of the lifting airscrew. Annalen der Physik, No. 626, 1937.

[3] Knight M. et Hefner R.A. Analysis of ground effect on the lifting airscrew. National Advisory Committee for Aero.

[4] Zbrozek J. Ground effect on the lifting rotor. Aeronautical Research Council, R M 2347, Juillet 1947.

[5] Cheeseman I.C. et Bennett W.E. The effect of the ground on a helicopter rotor in forward flight. Aeronautical Research Council, R M 3021, 1957.

[6] Hayden J.S. The effect of the ground on helicopter hovering power required. 32nd Annual

Forum of the American Helicopter Society, Washington D.C, Mai 1976.

[7] M.R Ritter. CFD analyses of the ground effect and the propulsion system of the BR2C micro drone. PhD thesis, Technical University of Munich, September 2008.

[8] V.K. Lakshminarayan et J.D. Baeder T.S Kalra. Cfd validation of micro hovering rotor in ground ef effect.

[9] V.J. Rossow. Thrust changes induced by ground and ceiling planes on a rotor in hover.

[10] H. Chung et K. Ryan D.C. Robinson. Computational investigation of micro rotorcraft near-wall hovering aerodynamics. International Conference on Unmanned Aircraft Systems (ICUAS), May 27-30 2014.

[11] C. Phillips. Computational Study of Rotorcraft Aerodynamics in Ground Effect and Brownout. PhD thesis, University of Glasgow, Mai 2010.

[12] N.D. Nathan. The Rotor Wake in Ground Effect and its Investigation in a Wind Tunnel. PhD thesis, University of Glasgow, Mai 2010.

[13] V.J. Rossow. Effect of ground and/or ceiling planes on thrust of rotors in hover. NASA Technical Memorandum 86754, Juillet 1985.

[14] Cristian Garcia Magana. Archean aerodynamics of rotor in confined environment. Institute for Space and Aeronautics, Juin 2015.

[15] V.K. Lakshminarayan. Computational Investigation of Micro-Scale Coaxial Rotor Aerodynamics in Hover. PhD thesis, University of Maryland, 2009.

[16] T. Jardin, S. Prothin, C. Garcia Magana. Aerodynamics of micro-rotors in confined environments. Journal of American Helicopter Society. Vol. 62, Number 2, pp. 1-7(7). 2017.

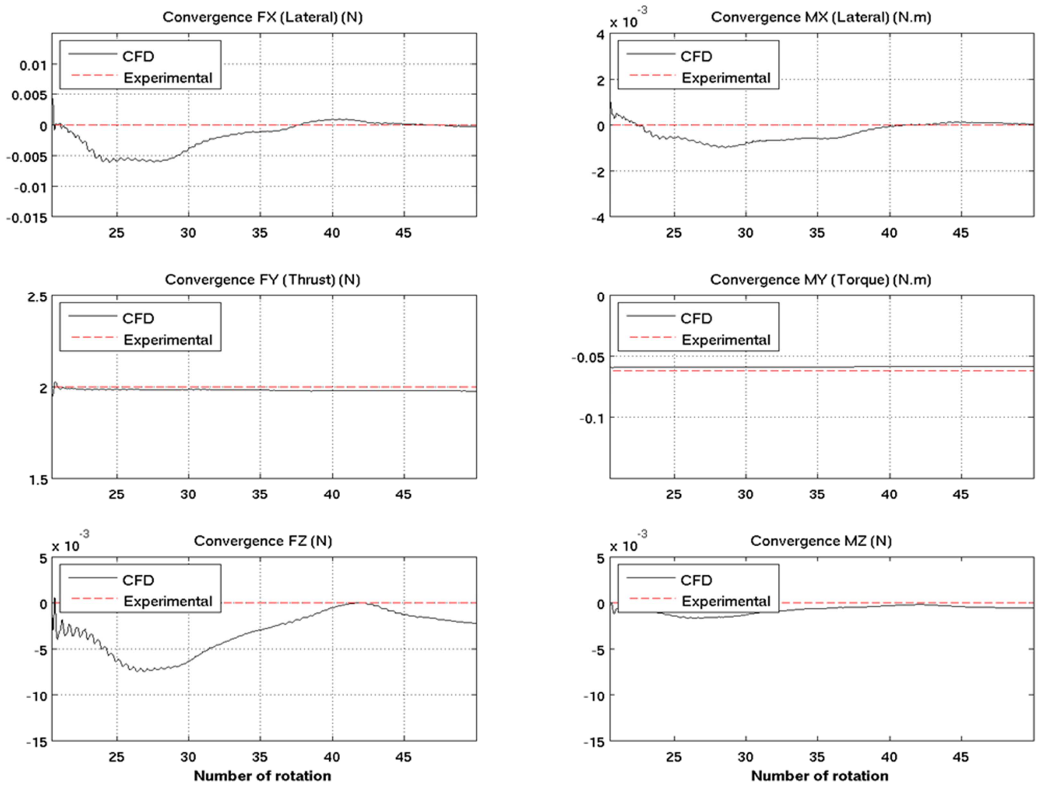


Figure 9 – Example of convergence in a simulation of a IGE case

	OGE		
	Exp	**Num**	**Ecart %**
RPM	3810,7	3810,7	0
T (N)	2,000	1,869	7%
Q (N.m)	0,058	0,059	-2%
PL	0,087	0,080	8%

Table 2 – Numerical and experimental comparison in OGE

IGE												
h/R	**RPM**	**T (N)**			**Q (N.m)**			**PL**			**Nb(num)**	
		Exp	**Num**	**Ecart %**	**Exp**	**Num**	**Ecart %**	**Exp**	**Num**	**Ecart %**	**Calculé**	**Exporté à partir**
0,25	3324,7	2,00	2,052	-3%	0,052	0,053	-1%	0,110	0,112	-2%	50	40
0,5	3465,0	2,00	1,888	6%	0,051	0,052	-2%	0,108	0,100	7%	50	40
1	3765,8	2,00	1,990	1%	0,062	0,059	5%	0,082	0,086	-4%	50	40
2	3779,4	2,00	1,864	7%	0,057	0,058	-2%	0,089	0,082	8%	50	40

Table 3 – Numerical and experimental comparison in IGE

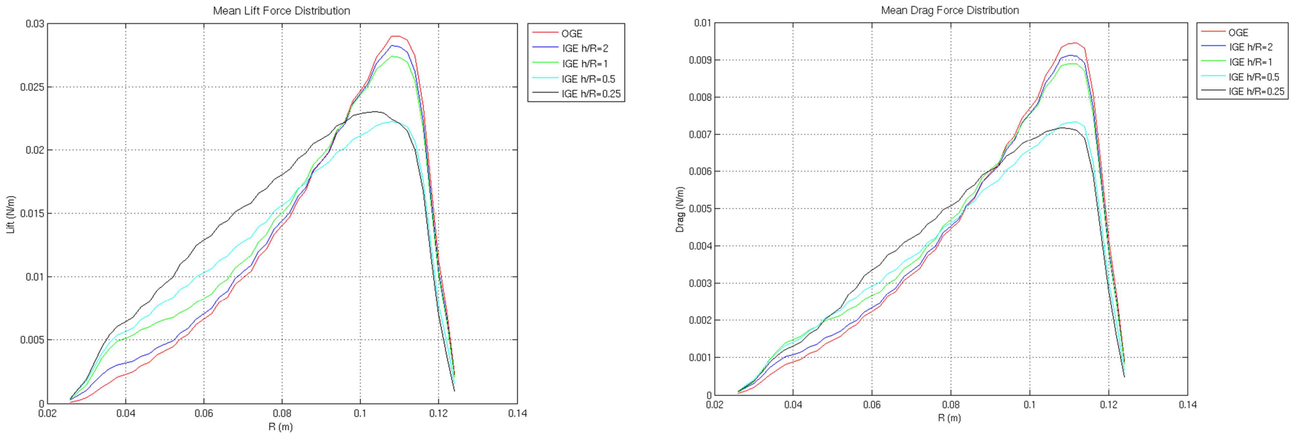


Figure 10 – Mean of lift/drag distribution for different IGE Cases (Numeical)

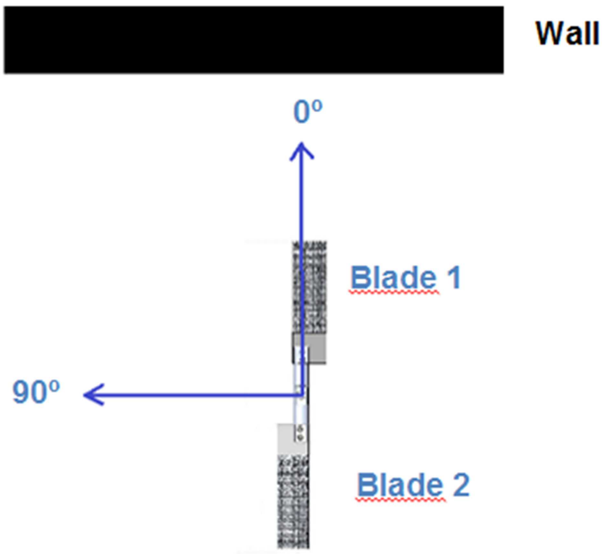


Figure 11 – ILoCE Diagram

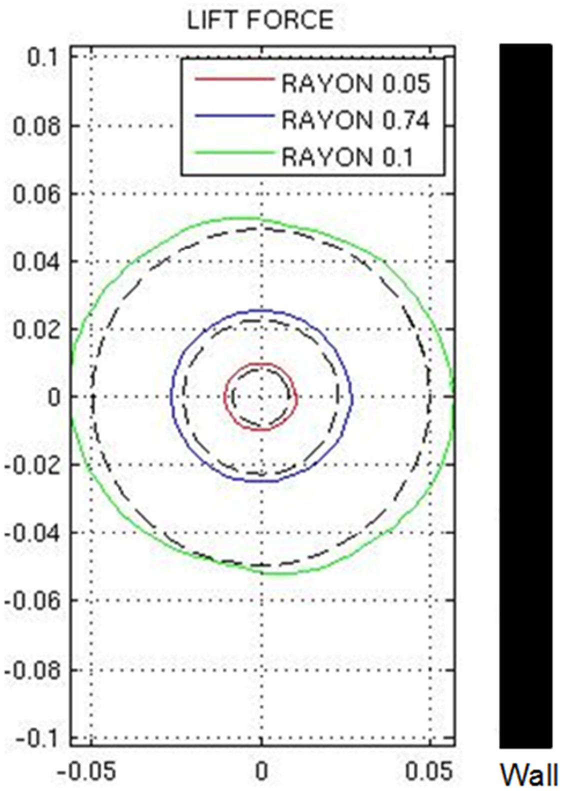


Figure 13 – Lift Distribution ILoCE h/R=2 d/R=2 (Numerical)

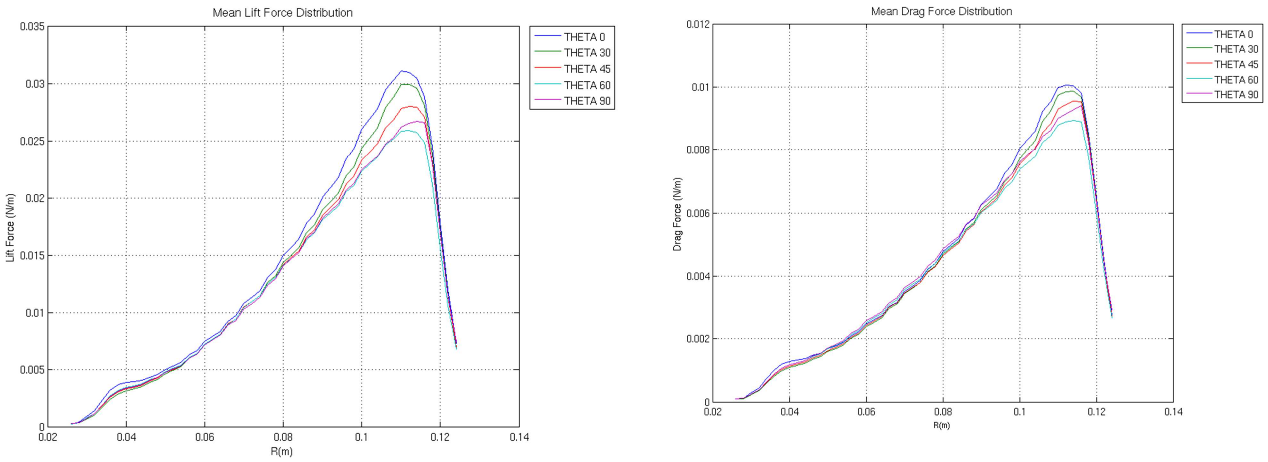


Figure 12 – Mean of lift/drag distribution for different ILoCE Cases (Numerical)

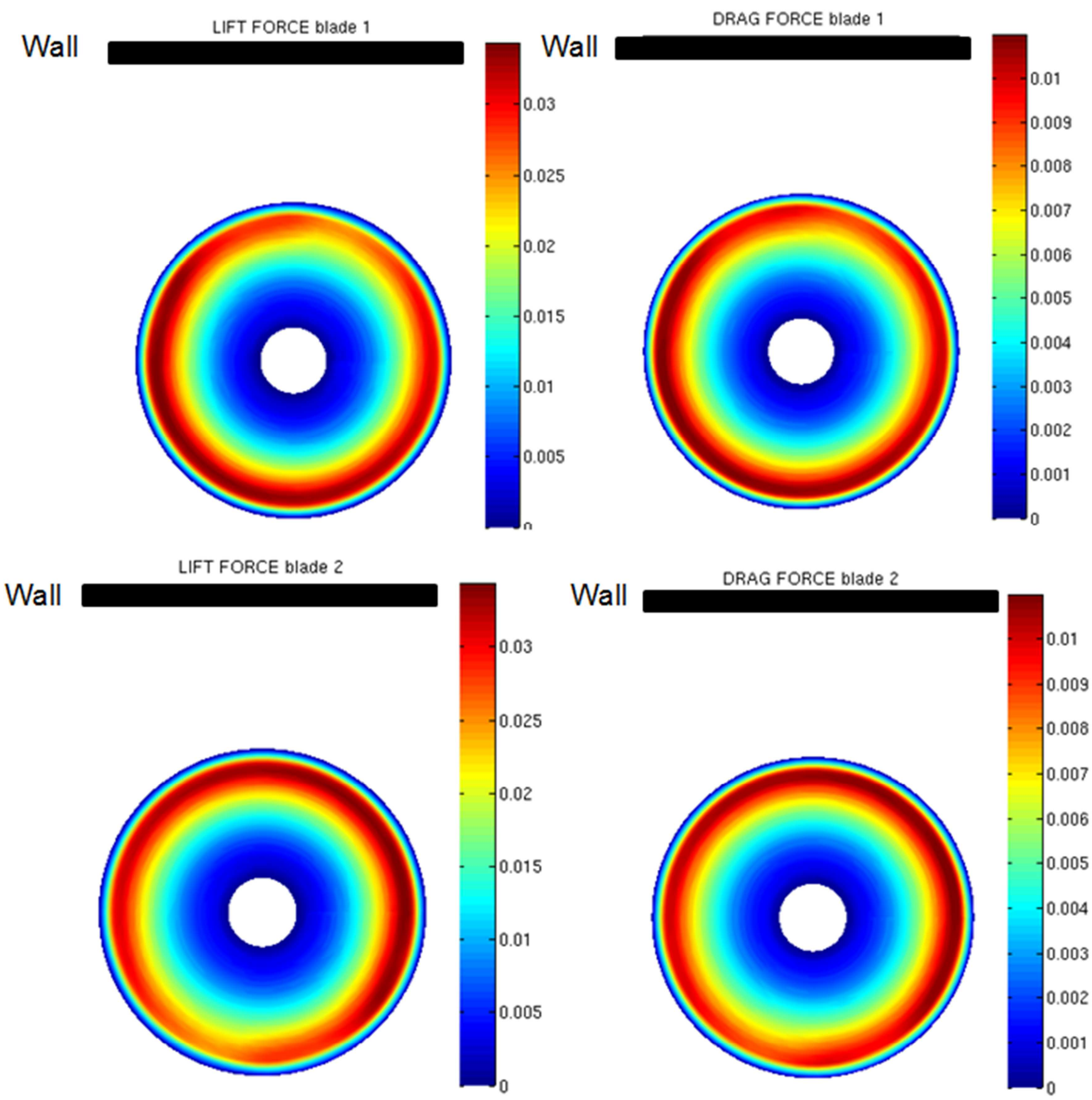


Figure 14 – Lift (left column) and Drag (Right column) force representation for the blade 1 (Upper Line) and blade 2 (lower line) (Numerical)

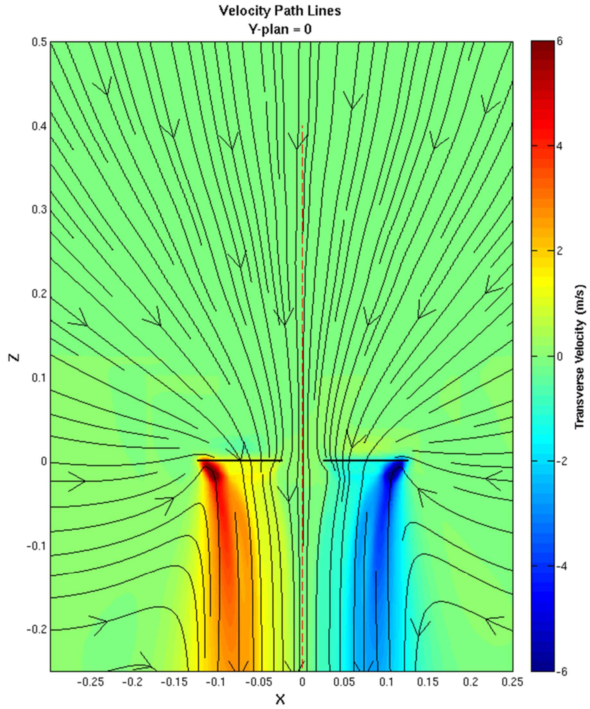


Figure 15 – Mean Velocity fields
OGE (Numerical)

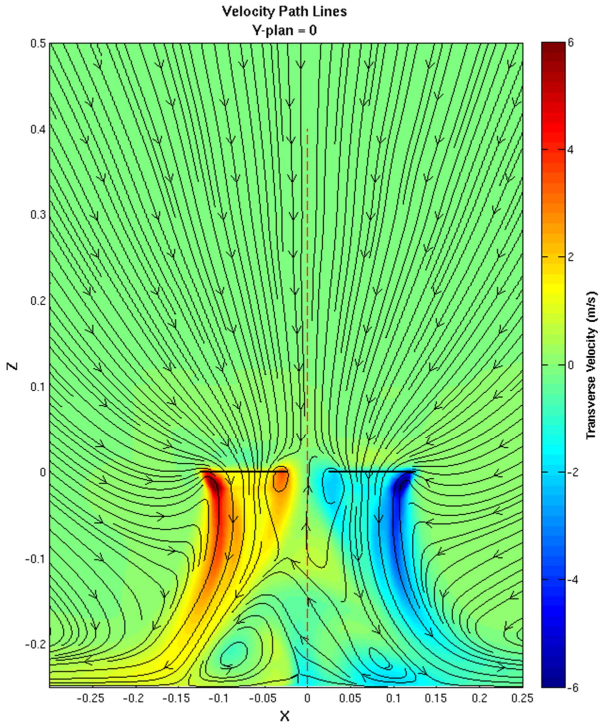


Figure 16 – Mean Velocity fields
IGE – h/R = 2 (Numerical)

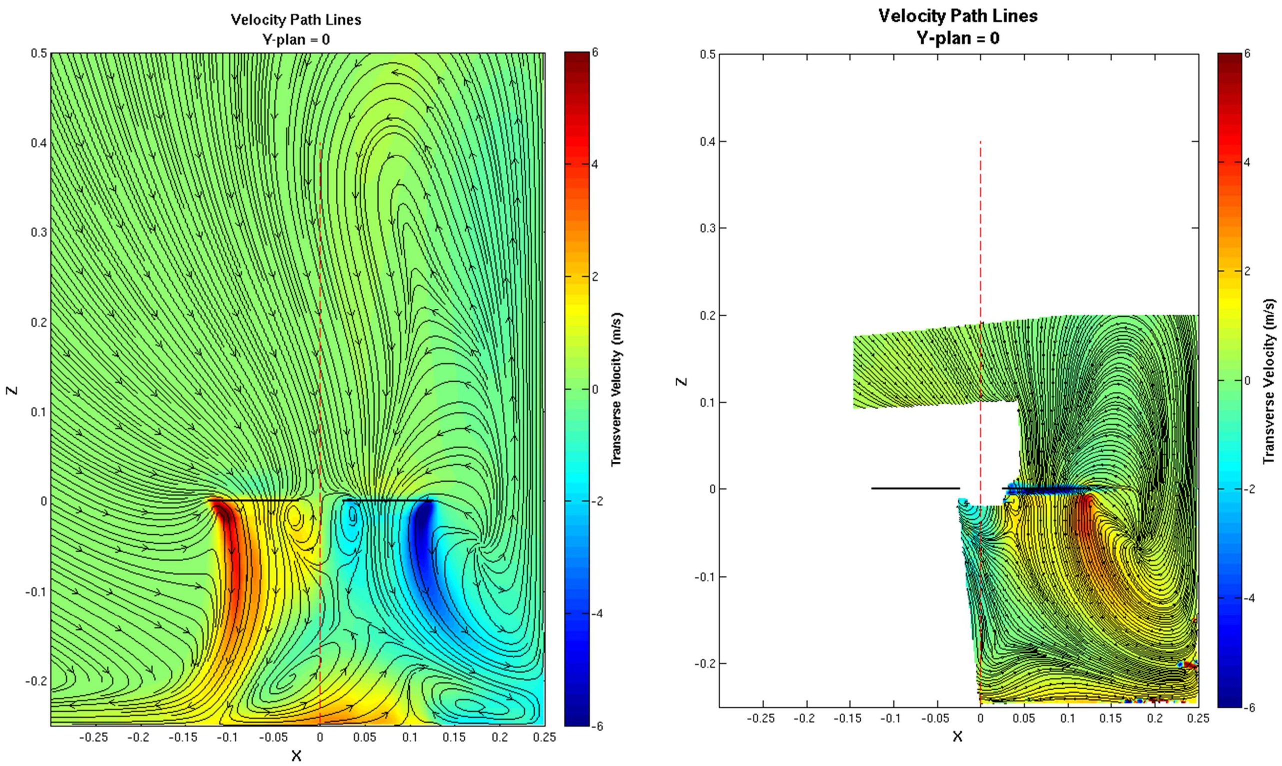


Figure 17 – Mean Velocity fields
ILoCE – h/R = 2 – d/R = 2
(Numerical : Left / Experimental : Right)

Investigation into the behaviour of an iced low Reynolds number aerofoil

N.L.Oo*, N.J.Kay, A.J.Brenkley and Dr R.N.Sharma
University of Auckland, Khyber Pass Rd, Auckland, New Zealand

ABSTRACT

Icing is a known hazard and has contributed to air accidents through its increased weight and reduced aerodynamic performance. A small UAV is more vulnerable to icing as it cannot afford the mass and power of de-icing systems, and, due to their small size, the added mass is proportionally greater. Compounding this are the effects of low Reynolds number flows.

To determine the impact of ice on a small UAV, a wind tunnel study was conducted with four ice shapes attached to the wing of a Kahu UAV at a Reynolds number of 200,000. Computational Fluid Dynamics (CFD) simulations to determine the change in stall behaviour. In all the cases, the ice accretion increased drag considerably, but the lift coefficient was altered for only one bluff-formed ice shape.

1 INTRODUCTION

Wing icing and the associated performance degradation is a known hazard for manned aircraft, being highlighted as the primary cause of notable crashes such as the loss of an ATR-72-212 over Roselawn, Illinois [1]. In flight wing icing forms as a result of supercooled water droplets in the air contacting the cold wing surface and freezing [2]. The form in which the ice freezes depends not only on the air temperature, but the amount of moisture in the air, the size of the water droplets, the relative speed of the wing, the wing leading edge radius and angle of attack.

Ice can form as one of three types. Rime is white and rough, while, glaze is clear, solid and smooth. Mixed consists of a combination of both. While rime ice remains close to the aerofoil form, glaze may form horns at the leading edge [3], significantly altering the aerofoil form. This change in geometry alters the associated forces and moments. Additionally, a significant mass is added to the airframe, increasing the demand on both engine and aerofoil and increasing surface roughness increases drag.

To counter this, manned aircraft designed to operate in potential icing conditions must be certified with appropriate de-icing equipment. For a small UAV, this is not a viable option. A small UAV is typically hand-launched, below 5kg, and so cannot afford the weight or power penalty associated with de-icing methods. Furthermore, the mass of the ice is proportionally more of the airframe weight [4] due to the low initial mass. The required excess power to counter the added mass may result in a much reduced endurance, or potentially result in loss of the aircraft.

Compounding this is the fact that small UAVs fly at low Reynolds numbers, typically below 250,000, due to their small size and low speed. This flow regime is dominated by a laminar and transitional boundary layer, with performance highly dependent on the behaviour of the Laminar Separation Bubble (LSB). This is a region of separated and recirculating flow, where the laminar boundary layer has detached from the aerofoil surface, but is able to reattach via transitioning to a turbulent flow and the enhanced momentum entrainment. The size of this bubble strongly influences lift and drag, and

*contact email: noo122@aucklanduni.ac.nz

depends on many parameters, but most importantly on the Reynolds number, Adverse Pressure Gradient (APG), upstream flow disturbances and surface roughness [5]. Increasing the Reynolds number, roughness and freestream disturbances will typically decrease the LSB size by accelerating transition, while increasing the APG provides the opposite effect. Hence, the addition of roughness due to ice may even aid aerofoil lift performance due to the reduced LSB size, at the cost of increased drag due to higher friction.

Prior emphasis has been on avoiding potential icing conditions [6], significantly limiting the operational utility of small UAVs. In order to assess the effect of ice on a small UAV in cruising flight, sampled ice shapes were mounted to the leading edge of a small UAV wing and physically tested. The aim was to assess the changes in the steady-state lift, drag and pitch moment coefficients at cruise. Additionally, CFD simulations were undertaken to provide more detailed flow assessment and visualisation for key test cases, to understand the changes in aerodynamic behaviour.

2 METHODOLOGY

2.1 Physical Testing

The initial phase of this study was conducted using the wing from a small Kahu UAV. Kahu is a fixed-wing UAV with a 2 m wingspan, which flies at approximately 60 km h^{-1}, or a Reynolds number of 230,000. This wing was mounted on the six-axis force balance of the Twisted Flow Wind Tunnel (TFWT) at the University of Auckland (UoA).

It was desired to assess the change in the performance of the wing with differing ice accretion. Four ice shapes were provided by the National Research Council of Canada (NRC) [7, 8]. These were generated in the NRC High-Altitude Icing Wind Tunnel to the standard of FAR25 Appendix C [9], simulating a variety of icing conditions and formation Angles of Attack (AoA). The ice shapes were replicated via 3D-Printed using Fused-Deposition Modelling. The ice shapes generated by NRC displayed similar roughness to that seen by Shin [10], and hence further roughening was not employed.

These cross-sections are presented in Figure 1, with the red representing the outer ice form, and the blue the aerofoil leading edge. From left to right, the ice shapes are designated by their formation AoA and ice formation condition, where IM is Intermittent Maximum, equivalent to Cumuliform clouds at -20°C, while CM is Continuous Maximum, representing Stratiform clouds at -5°C.

Figure 1 – Ice Shape Cross-Sections [7, 8]

Testing was conducted at a Reynolds number of 200,000 based on the mean wing chord. The freestream turbulence intensity was measured to be 1.1%, with a floor boundary layer depth of 400 mm. For this reason, the wing was mounted in the vertical orientation 520mm above the floor level, a splitter plate providing a reference plane at the wing root. Data was gathered from -4° to 20°, covering typical flight attitudes.

Data was gathered solely through the force balance, as the wing could not be modified. The force and torque loads of the mount were measured and subtracted from all the data. The force balance resolution was 0.05 N for forces and 0.1 Nm for moments [11]. As the force resolution was over an order of magnitude lower than the lowest drag coefficient anticipated for the clean aerofoil per the data of Selig et al. [12], this was sufficient for the measurement of this data. The pitch moment resolution, however, was comparable to the data expected, and so can only be seen as a reference trend. As the force balance is not configured for high-frequency acquisition, time-averaged coefficients were derived.

2.2 CFD

The stall behaviour of the clean aerofoil and the 0° IM ice shape were investigated further via CFD. The turbulence model employed was Scale-Adaptive Simulation-Shear-Stress Transport (SAS-SST). This is an Unsteady Reynolds-Averaged Navier-Stokes (URANS) model, based on the SST formulation for low-cost solving where the flow is stable. However, unlike basic SST, SAS-SST adjusts its source terms, allowing it to simulate large-scale flow structures, such as those in the wake of a stalled aerofoil, in a manner comparable to Large Eddy Simulation (LES) [13]. The capability of the SAS-SST in the wake region was proven by Garbaruk et al. [14], with the numerical results comparable to the experiment.

A C-grid configuration was used, maintaining an inlet three chord lengths upstream of the clean wing leading edge, and an outlet opening five chords downstream. This was based on a prior LES study on aerofoils in similar flow conditions [15]. Symmetry planes were used to enforce a two-dimensional set-up to focus on the aerofoil behaviour in isolation of wing effects. Free-slip wall boundary conditions at the top and bottom of the domain were kept at a reasonable distance, so as not to have any influence on the solution.

The compact domain allowed for a high mesh density in proximity to the wing and its wake, as shown in Figure 2. The meshing was completed with a total of 383,298 elements using ANSYS ICEM-CFD. The wall normal mesh was kept at $\Delta y^{+} < 1$ and the chordwise mesh resolution of $\Delta x^{+} = 10$ over the ice accreted chord and $\Delta x^{+} = 28$ at the downstream of the ice-shape. Although not as sensitive as LES, Egorov and Menter [16] demonstrated that energy dissipation is dependent on grid size, and so the use of an LES grid provided suitable resolution in the wake.

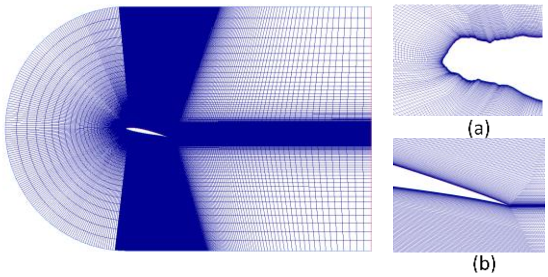


Figure 2 – 0° IM Aerofoil mesh at 14.5° AoA, with detail (a) leading-edge (b) trailing-edge

Simulations were run for five seconds with 5,000 time steps. The same time step of 0.001s, was set to the same for all of the simulations, enabling aerodynamic unsteadiness to be observed up to the maximum frequency of interest of 500 Hz.

3 RESULTS

3.1 Physical Testing

Figures 3 and 4 compare the aerodynamic performance of the test wing with each ice form and the baseline clean wing. Evident in Figure 3 is that the ice has no apparent impact on the lift coefficient below 10° AoA. While the form of the aerofoil is altered, reducing camber at the leading edge, this effect is countered by the added surface area, producing the same lift coefficient for the given reference clean wing planform.

Beyond 11° AoA, however, the ice shapes produce differing behaviour. The 0° IM ice shape most notably results in an earlier stall, with the corresponding reduced maximum lift coefficient. This is the most bluff-formed ice shape with a more sudden change in form. However, it also has the gentlest stall, suggesting a different stall mode from the other forms. In terms of flight operation, this early stall would suggest reduced AoA limits should be enforced when icing is present to maintain safe flight.

By comparison, the clean wing has the sharpest stall, with a gentle reduction followed by a sharp drop. This suggests a trailing-edge stall reaching the LSB and commencing full separation. The remaining three ice shapes are similar in performance, with a higher maximum lift coefficient and later stall, which is slightly gentler than the clean aerofoil case. This is likely a result of flow instabilities generated by the ice, resulting in a higher energy boundary layer which thus remains attached longer.

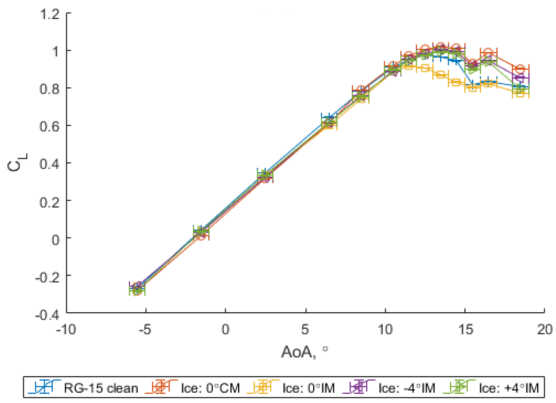


Figure 3 – Lift Coefficient against AoA

Similarly, Figure 4 shows an increased drag coefficient for the 0° IM ice shape for a given lift coefficient, with the exception of at lift coefficients between 0.6 and 0.8. In this region, the -4° IM ice shape has the highest drag. This is because the ice sits entirely on the upper surface of the leading edge, with its depth more exposed as AoA increases. As expected, the clean wing has the lowest drag at all AoA, and thus the lowest power requirements.

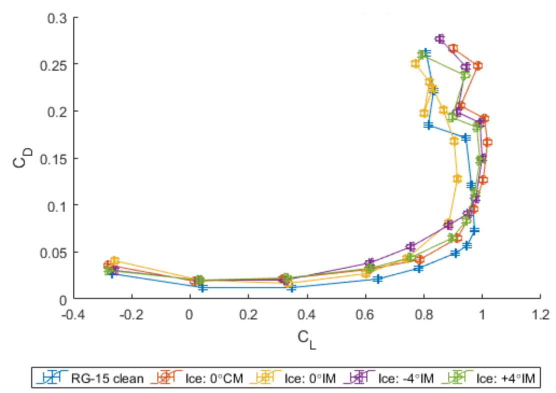


Figure 4 – Drag Coefficient against Lift Coefficient

Of note is that, for this UAV, the mass of the ice was estimated at 0.2 kg, approximately 10% of the UAV mass. This requires the AoA to be raised by 1°. Combined with the increased drag coefficient for a given AoA, this has a significant effect on endurance, potentially up to 50%.

3.2 Numerical Simulation

The CFD allows visual inspection of the difference in stall behaviour of the clean and 0° IM cases, and comparison of the pressure distributions. The selected AoA for assessment were 11.5° and 14.5°, corresponding to the stall AoA of the 0° IM and clean aerofoil, respectively. Figure 5 represents the velocity contours of clean and ice-accreted aerofoil at these AoA.

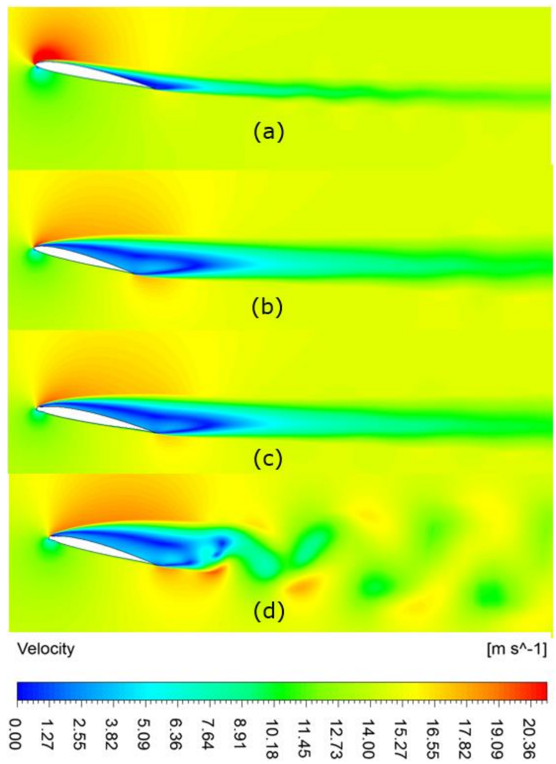


Figure 5 – Velocity contour plots (a) 11.5°, clean (b) 14.5°, clean (c) 11.5°, 0° IM (d) 14.5°, 0° IM

On the clean aerofoil, a short LSB was observed at around the leading-edge of the aerofoil at 11.5°, as seen in Figure 6. A large stagnant area is also seen towards the trailing edge in this condition in Figure 5. At 14.5°, this has reached the LSB and complete, but stable, separation ensues, which is the start of stall. In comparison, the bluff-edge 0° IM case has reached this state at the lower angle of 11.5°.

This is also reflected in Figure 7, with the 11.5° AoA, where the suction surface shows a similar C_p to that of the clean wing at 14.5°. In this case, the flow cannot reattach after separating on the blunt form of the ice. By 14.5°, the ice-accreted aerofoil shows the clearly defined vortex street seen in Figure 5, acting as a bluff-body separation. Of interest also are the small pressure pockets in the cavities of the ice on the pressure side.

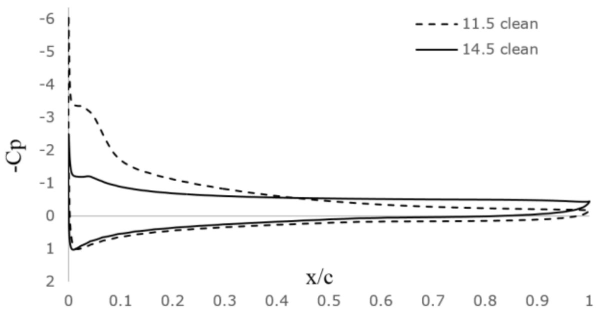


Figure 6 – Pressure coefficients comparison for clean aerofoil at 11.5° and 14.5°

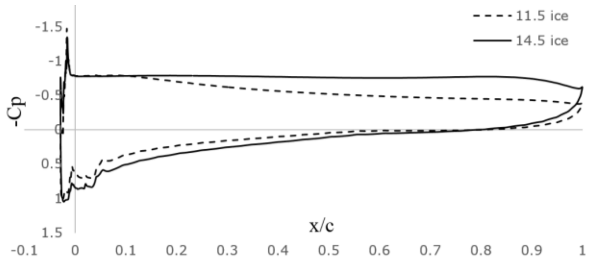


Figure 7 – Pressure coefficients comparison for ice-accreted aerofoil at 11.5° and 14.5°

Mirzaei et al. [17] observed that the fundamental vortex shedding frequency for ice-accreted aerofoil decreases with an increase in AoA, similar to a clean aerofoil. This was also observed in the CFD results, as shown in Figures 8 and 9, in which the velocity Power Spectral Densities (PSD) at a point in the wake, 2.5 chords downstream, are plotted. The fundamental frequency decreases from 69 Hz at 11.5° AoA to 55.6 Hz at 14.5° for the clean aerofoil. In the ice-accreted cases, however, the fundamental and secondary peaks were notably more prominent, while being only 4 Hz and 2 Hz lower for 11.5° and 14.5° AoA, respectively. This is a result of the greater vorticial energy provided by the rough ice form, maintaining time-averaged suction pressures and thus the gentler stall. In all the simulated cases, using the projected flow-normal aerofoil depth, the Strouhal numbers are in good agreement with that expected from bluff-body vortex shedding [18], being between 0.21 and 022.

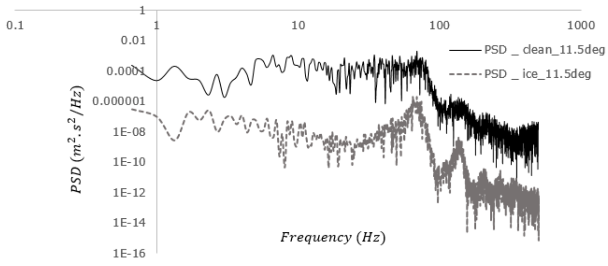


Figure 8 – Velocity PSD comparisons for clean and ice accreted aerofoil, 11.5° AoA. 0° IM is decreased by 3 orders of magnitude for clarity

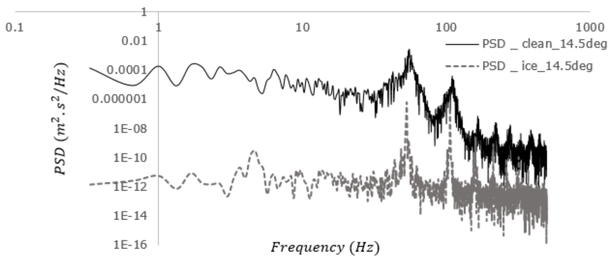


Figure 9 – Velocity PSD comparisons for clean and ice accreted aerofoil, 14.5° AoA. 0° IM is decreased by 6 orders of magnitude for clarity

4 CONCLUSIONS

The wind tunnel study showed that ice accretion does have a significant effect on UAV operation, primarily due to the increase in drag and so the corresponding loss of endurance. This would need to be accounted for in mission planning. The lift coefficient was only altered in one case, reducing the stall AoA to 11.5°. As such, when flying in icing conditions, the AoA envelope permitted should be reduced to account for a reduced stall tolerance, but lift capability is unaltered below this.

The CFD shows this change is attributed to the bluff form of the ice, which does not permit flow reattachment and hence no LSB is formed. The amplitude of fluctuations is also greater, necessitating greater control response.

While the CFD in this study examined the high-AoA cases, given the increased fluctuation with the ice, it would be desirable to investigate fluctuations in the forces acting on the wing at low AoA, in normal cruise. This would assist in adapting control systems to be more robust when ice may be present. Furthermore, the effects of span should also be assessed, as turbulence is fundamentally three-dimensional.

ACKNOWLEDGEMENTS

The authors would like to extend their thanks to the NRC and the Defence Technology Agency of the New Zealand Defence Force for providing the physical test specimens, and the New Zealand eScience Infrastructure for use of their high-performance cluster for the simulations.

REFERENCES

[1] NTSB, "Aircraft Accident Report: In-flight Icing Encounter and Loss of Control," NTSB, NTSB/AAR-96, Washington D.C, 1994.

[2] National Weather Service, "Aircraft Icing," http://www.crh.noaa.gov/Image/lmk/Brian%20S/LMK_Icing_Show.pdf [cited December 17 2014].

[3] FAA, "Chapter 15: Ice and Rain Protection," https://www.faa.gov/regulations_policies/handbooks_manuals/aircraft/amt_airframe_handbook/media/ama_Ch15.pdf [cited March 25 2015].

[4] Szilder, K., and McIlwain, S., "In-flight icing of UAVs-the influence of flight speed coupled with chord size," *Canadian Aeronautics and Space Journal,* Vol. 58, No. 2, 2012, pp. 83-94.

[5] Gaster, M., "The structure and behaviour of laminar separation bubbles," Aeronautical Research Council, R&M No. 3595, London, UK, 1969.

[6] Siquig, R.,A., "Impact of Icing on Unmanned Aerial Vehicle (UAV) Operations," Naval Environmental Prediction Research Facility, PR 90:015:442, 1990.

[7] Brenkley, A., "Aerodynamic Performance of the Kahu UAV with Ice Formation on the wings at Low Reynolds Numbers," Part IV project report, 2015-ME14, 2015.

[8] Kay, N.J., "Aerodynamic Performance of the Kahu UAV with Ice Formation on the wings at Low Reynolds Numbers," University of Auckland, ME053-2015, Auckland, New Zealand, 2015.

[9] FAA, "Appendix C to Part 25," http://www.ecfr.gov/cgi-bin/text-idx?rgn=div5;node=14%3A1.0.1.3.11#ap14.1.25_11801.c [cited 19 September 2015].

[10] Shin, J., "Characteristics of Surface Roughness Associated With Leading Edge Ice Accretion," *AIAA 32nd Aerospace Sciences Meeting and Exhibit,* Vol. NASA TM-106459, AIAA, Reno, Nevada, 1994,

[11] Yacht Research Unit, "TFWT Specifications," http://homepages.engineering.auckland.ac.nz/~dpel004/yru/html/Wind-tunnel/ [cited March 21 2015].

[12] Selig, M.S., Guglielmo, J.J., Broeren, A.P., "Summary of Low-Speed Airfoil Data, Volume 1," SoarTech Publications, Virginia Beach, Virginia, USA, 1995,

[13] Menter, F.R., "Best practice: scale-resolving simulations in ANSYS CFD," *ANSYS Germany GmbH,* 2012, pp. 1-70.

[14] Garbaruk, A., Shur, M., Strelets, M., "NACA0021 at 60 deg. incidence," *Notes on Numerical Fluid Mechanics and Multidisiplinary Design,* Vol. 103, 2009, pp. 127-139.

[15] Itsariyapinyo, P., and Sharma, R., "NACA0015 Circulation Control Airfoil Using Synthetic Jets at Low Angles of Attack and Low Reynolds Number," *8th AIAA Flow Control Conference,* 2016, pp. 3772.

[16] Egorov, Y., and Menter, F., "Development and application of SST-SAS turbulence model in the DESIDER project," *Advances in Hybrid RANS-LES Modelling,* Springer, 2008, pp. 261-270.

[17] Mirzaei, M., Ardekani, M.A., and Doosttalab, M., "Numerical and experimental study of flow field characteristics of an iced airfoil," *Aerospace Science and Technology,* Vol. 13, No. 6, 2009, pp. 267-276.

[18] Yarusevych, S., Sullivan, P.E., and Kawall, J.G., "On vortex shedding from an airfoil in low-Reynolds-number flows," *Journal of Fluid Mechanics,* Vol. 632, 2009, pp. 245-271.

An Adaptable Indoor Flight Test Implementation for small UAVs

Pengfei Yu*, Derrick Ho, and K.C Wong
The University of Sydney, Aeronautical Engineering Building, Sydney

ABSTRACT

In this paper, an adaptable indoor flight test implementation for small unmanned aerial vehicles (UAVs) with motion capture system (Mocap) is illustrated. By generating pulse position modulation (PPM) signals which conform to multi-protocol compatible radio control (RC) module, it enables control of commonly available commercial RC products which usually do not provide application program interface (API) to users. A PPM signal reader which receiving the exact same signal as the onboard receiver is constructed into the system to measure the signal latency, control signal loss and to retrieve the trim value for different manoeuvres of any specific aircraft. The implementation and results of commanded trajectories (circle, figure ”8” and parabolic paths) were tested to explore the viability and adaptability of the presented method.

1 INTRODUCTION

Indoor testbeds for UAVs began to emerge nearly 12 years ago[1]. Progress in research has also excelled with the provision of Mocap system that provides highly accurate position and attitude feedback towards indoor testbeds such as aerial manipulation[2]–[5], aerobatics[6,7,8], coordinate construction[9,10] and others. By using the a Mocap system such as Vicon or Optitrack to provide accurate state space estimation for control, researchers can focus on higher level activity due to the control algorithm being separated from state estimation. Researchers could get submillimeter accuracy with Mocap real-time streaming information even not fused with onboard sensors[11]. Examples of research approaches using Mocap include RAVEN from MIT[1], and the Flying Machine Arena from ETH[12]. Our test implementation has similar architecture on top level with the abovementioned setups. We extend this system architecture to adapt to both customized research platforms and commercial products which commonly use RC gear for control input. While the cost of Mocap systems is relatively high, it allows potential low-cost research on robot collaboration and swarm studies due to the low cost of commercial aerial platforms that can be as low as A$22[13].

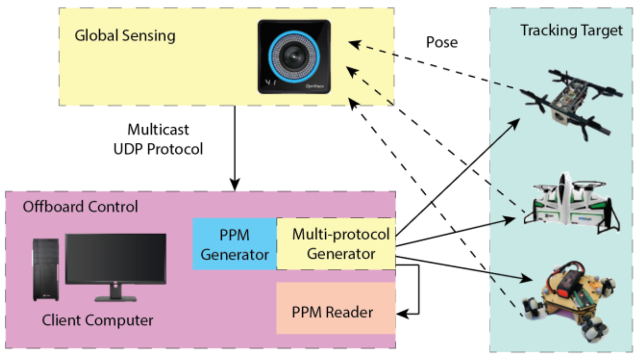


Figure 1- System Overview

By developing a PPM signal generator and reader, commercial products which do not offer APIs can be controlled using our implementation. By utilising the PPM signal reader, control output and input can be computed in a closed loop feedback system as seen in Figure 1. Researchers can then decipher the required PPM output towards controlling the commercial products, measuring

*contact email: peyu7686@uni.sydney.edu.au

control command latency, loss of the command and get trim value of a specific state. With this methodology, the overall outcome allows researchers the ability to control a broad range of platforms and also reverse engineering the details with the PPM signals measuring.

This paper is organised as follows: Section 2 presents overview of system setup and performance. Section 3 shows test result from our customized research platform and commercial products. Section 4 provides summary and future work.

2 SYSTEM OVERVIEWS

Mocap systems provide global sensing of the objects position and orientation and offboard computer offer extra computational power to process the information; typically with high computational control algorithm to traverse mobile platforms. The general data flow of our implementation is illustrated in Figure 1: pose of target is estimated by software running on a host machine from Mocap system. This host machine broadcasts the estimation through UDP protocol and any client machine in the same network with host machine can access these state estimations simultaneously with other clients and use it in their own applications such as specific control algorithms. Control commands are generated through a PPM signal generator to a multi-protocol transmitter module. For typical usage, command signals would be sent to the receiver on target UAVs and a separated receiver connected to a PPM signal reader which send back the actual received command information to offboard computer to calculate command latency, command loss and get a trim value of specific state.

2.1 Hardware for global sensing

Hardware includes two parts. Part one is for global sensing and part two is for offboard control.

The hardware for global sensing in our setup includes the following in Table 1:

Component	Part	Detail
Camera	Optirack 41W	4.1MP Horizontal FOV 51°
	Optirack 17W	1.7MP Horizontal FOV 70°
Host	Computer Windows OS	CPU: Intel CoreI7-6700K NIC: Intel E1G42ETBLK Dual Port Adapter
Network	Wi-Fi Router	TY-LINK ARCHER AR2600
Software	Motive 2.0	

Table 1 - Hardware for Global Sensing

The Mocap system software (Motive) is running on Windows operating system. The host machine is connected to a WIFI router which could provide a stable network connection to client machine through WIFI or Ethernet cable. The system is capable of transmitting information at the rate of over 200Hz, however typical 180Hz is used.

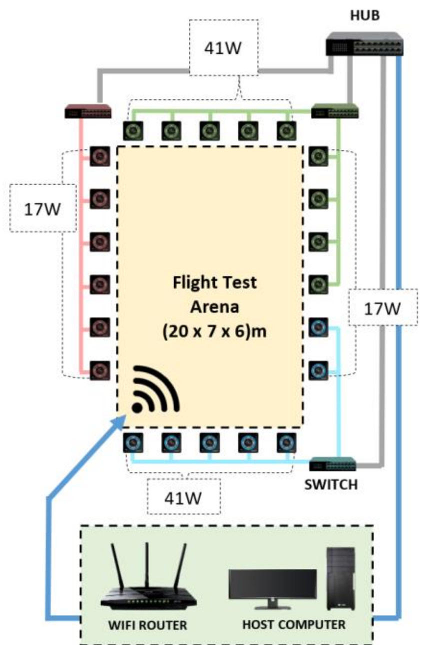


Figure 2 - Hardware for global sensing

A rigid body consisting of 3 or more reflective markers allows tracking of position and orientation. For single marker target, only position can be tracked. All estimated state is relative to user defined coordinate. The latency from the Optitrack software to achieve global sensing is approximately 5ms. The UAV-Lab flight test arena offers a 140m^2 (20m×7m×6m) useable volume which could potentially be utilized for flight testing small and slow fixed-wing or small vertical take-off and landing (VTOL) aircraft in fixed-wing mode. With this setup, the effective detectable range for a maker size of 3-4 cm is approximately 23m and therefore the whole volume is capable and effective towards detecting the markers. Increasing the brightness threshold for marker detection yields higher reliability due to eliminating sources of low reflectivity (or otherwise noise towards markers due to unwanted reflections).

2.2 Hardware for offboard control

Our hardware for offboard control include a PPM signal generator, a multi-protocol RC module (Figure 3) and a PPM signal reader shown in Figure 5. A PPM signal is generated conforming to the multi-protocol RC module. This allows the user to manipulate and orchestrate the output of the UAV by constructing the correct length of pulses within the PPM signal. In lieu of this, and in conjunction with the multi-protocol transmitter module, this ideology can therefore adapt to DSMX, DSM2 (Spectrum), D8, AFHDS (Frsky), FASST (Futaba), A-FHSS (Hitec) as popular RC-protocol examples. As it includes most main stream protocols provided on the current market, which makes it adaptable to a majority of the off-the-shelf product which allows a broad selection of platforms to be acquired and tested. For instance, we could adapt to control mini quadrotor (Eachine E010[13]) costing A$20 to higher calibre VTOL UAV aircraft such as a X-Vert or any other commercial ones, provided it uses a common RC protocol. Noteworthy when using multiprotocol modules, it needs to bind to the same protocol (a compatible) receiver in the traditional RC way; typically, through a transmitter that supports external radio modules such as a FrSky Taranis XD9 as one example.

Figure 3 - Multi-protocol Module[14]

In a PPM signal frame (Figure 4), each channel maps a certain controller output from 1000 (1ms) to 2000 (2ms). Each channel is separated by a small-time gap (0.4ms). The total frame length of the signal is 22.5ms for an 8 channels PPM command.

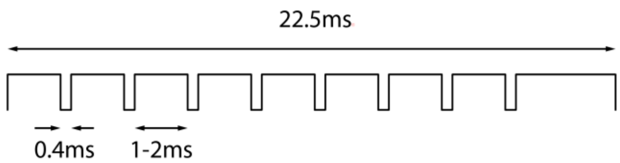


Figure 4 - PPM Signal (8 channels)

PPM reader assumes that two identical receivers which both bind to the same transmitter module receives the radio signal at the same time. Like the way that researchers construct satellite receiver configuration to have better access to the radio signal and provide extra redundancy in the system. Under this assumption, the PPM reader is employed to measure the latency of when the radio command is received between the transmitter and receiver. Similarly, the PPM reader can also be used to record the pilot controls during manual UAV flying operation from a transmitter by logging the PPM signal to determine the intended control inputs. It is also a helpful feature which could help to find out the trim value for certain state of the test aircraft; e.g. hover trim values of a VTOL UAV.

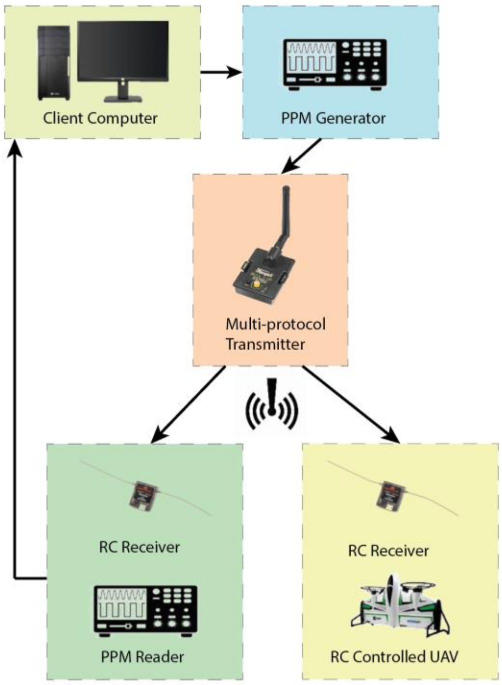


Figure 5 - Hardware for offboard control

2.3 Latency

System latency comes from mainly three sources:
(1) Mocap system capture latency;
(2) PPM signal generating time;
(3) radio signal decoding latency;

Figure 6 shows the system latency in average which can accumulate to 50ms on average based on methods described later. Note our method is one direction only, and it also has variable latency.

For small size UAV such as crazyflie which is also a 22g mini quadrotor, motor response time could be as high as 200-300ms[15], which is significantly higher than the aforementioned system latency. So, in term of real world response time, the dominated term is still the mechanical time constant.

In Figure 7, blue line is the PPM signal sent with its timestamp and red line is PPM signal received with its timestamp. X axis is system time, unit is second. Y axis is the sweep signal which ranges from 1000 to 1900.

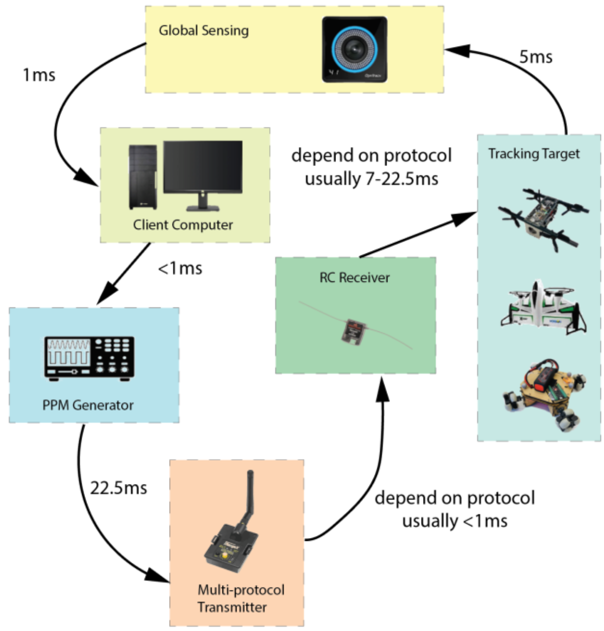


Figure 6 - System latency

The setup detail is that a PPM generator generates a known pattern of signal numbers from 1000 to 1900 to conform to typical RC pulse width ranges, which is then transmitted by the multiprotocol transmitter module. The RC receiver receives the signal numbers and log the time it collects the signal numbers and sends it back to user's computer. System latency could be measured by comparing the time difference between when it is transmitted and received.

The receiver relies on the protocol that is used. Certain protocol such as SBUS has significantly lower latency compared to other protocols.

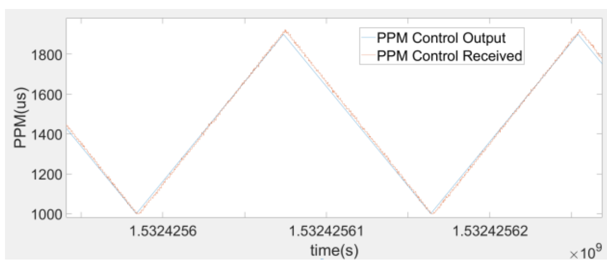


Figure 7-Measured latency

2.4 Logging and Failsafe

One benefit of using Multi-protocol module is that researchers could mount this module onto a commercial transmitter. By setting up the trainer port on a commercial transmitter, researchers could switch back to manual control whenever there is something unexpected happen via flipping trainer switch.

Note that if you use commercial transmitter as a failsafe method, dual-rate, exponential function and trim values should all be reset to avoid interfering and undesired adjustments of the PPM command that is intended to be transmitted.

In summary, all desired pose, actual pose, command sent, and command received are logged in the client computer which allows researchers to plot, analyse and replay all information for further analysis.

3 FLIGHT TESTS

3.1 Test platform

Two test platforms are chosen:(1) Mini 22g QX65 quadrotor by Eachine, and (2) A 210g fixed-wing VTOL aircraft capable of forward flight; X-Vert by E-flite.

The mini quadrotor allows basic testing due to its inherent stability provided by the manufacture while the fixed-wing UAV offers the ability to conduct forward flight testing in future aerodynamic flight control phases.

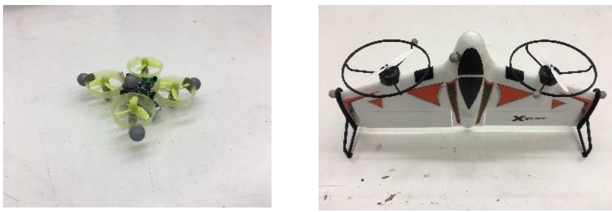


Figure 8 - QX65 and X-Vert with makers

3.2 Controller Diagram

Our controller follows traditional cascaded control methodology for multicopter (Figure 9). In our case, the onboard attitude controllers are employed for overall flight stabilisation. Customized PID position controller is running on offboard computer at 50Hz. The desired position of vehicle in world frame is r_t, actual position is r, and acceleration is $\ddot{r}$.

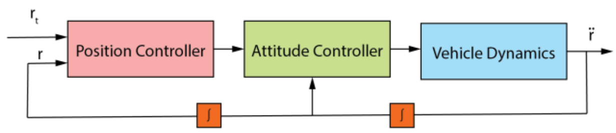


Figure 9 - Control Diagram

3.3 Experiment results

Vehicles are controlled to follow trajectories of a circle, a figure “8” and a parabola. Here are the results of these tests. These tests illustrate that current implementation could effectively track desired trajectory as they are shown in Figure 10 and 11.

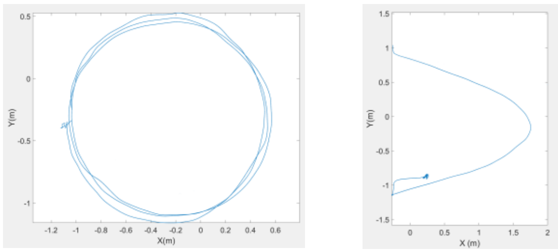


Figure 10 - Circle and Parabola

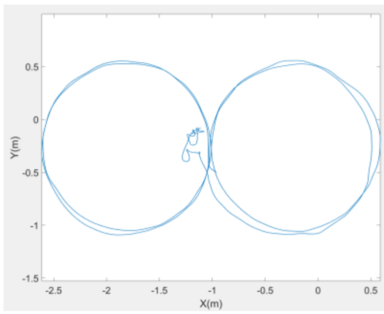


Figure 11 - Figure "8"

4 CONCLUSIONS AND FUTURE WORKS

In this paper, we have introduced a practical implementation for indoor UAV flight testing under closed-loop control with the assistance of Mocap system and multi-protocol modules. The benefit of this method is that it could adapt to a variety of platforms including not only customized but also commercial available RC UAV with the ability to quantify latency. By constructing a PPM signal reader into the system, users could also monitor the variable latency, loss of command signal and get a trim value for the specific state. Test video of this paper is available on YouTube: https://youtu.be/fV1l--NOZWM.

REFERENCES

[1] J. P. How *et al.*, "Real-time indoor autonomous vehicle test environment," *IEEE Control Syst. Mag.*, vol. 28, no. 2, pp. 51–64, Apr. 2008.

[2] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 2491–2498.

[3] F. Huber *et al.*, "First analysis and experiments in aerial manipulation using fully actuated redundant robot arm," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 3452–3457, 2013.

[4] A. Torre, D. Mengoli, R. Naldi, F. Forte, A. Macchelli, and L. Marconi, "A prototype of aerial manipulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2653–2654.

[5] A. Q. L. Keemink, M. Fumagalli, S. Stramigioli, and R. Carloni, "Mechanical design of a manipulation system for unmanned aerial vehicles," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 3147–3152.

[6] S. Lupashin *et al.*, "A simple learning strategy for high-speed quadrocopter multi-flips," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 1642–1648.

[7] M. Hehn and R. D'Andrea, "A flying inverted pendulum," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 763–770.

[8] D. Brescianini, M. Hehn, and R. D'Andrea, "Quadrocopter pole acrobatics," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3472–3479.

[9] F. Augugliaro, E. Zarfati, A. Mirjan, and R. D'Andrea, "Knot-tying with flying machines for aerial construction," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 5917–5922.

[10] F. Augugliaro *et al.*, "The flight assembled architecture installation: Cooperative contruction with flying machines," *IEEE Control Syst.*, vol. 34, no. 4, pp. 46–64, 2014.

[11] D. Mellinger, "Trajectory Generation and Control for Quadrotors," *PhD*, 2012.

[12] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The Flying Machine Arena," *Mechatronics*, vol. 24, no. 1, pp. 41–54, 2014.

[13] "Eachine E010." [Online]. Available: https://www.banggood.com/Eachine-E010-Mini-2_4G-4CH-6-Axis-Headless-Mode-RC-Quadcopter-RTF-p-1066972.html?rmmds=search&cur_warehouse=AU.

[14] "Multi-protocol Module." [Online]. Available: https://hobbyking.com/en_us/jumper-jp4in1-multi-protocal-radio-transmitter-module.html.

[15] Y. Chen and N. O. Perez-Arancibia, "Generation and real-time implementation of high-speed controlled maneuvers using an autonomous 19-gram quadrotor," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3204–3211.

Simulation & Flight Test of Ducted-fan UAVs for Formation Flight

Taegyun Kim, Hyemin Mo, Seungkeun Kim,* and Jinyoung Suk
Department of Aerospace Engineering, Chungnam National University, 99 Daehak-ro, Yuseong-gu, Daejeon 305-764, Korea

ABSTRACT

This paper proposes to drive a group of ducted-fan UAVs in a formation, using Leader-follower strategy for surveillance and scouting. To each UAV the target position is assigned based on the leader UAVs position. The UAVs computes their position commands using the relative position data between the UAVs in order to keep a formation. The results via ardupilot SITL verifies that the UAVs are able to keep a predetermined formation for mission accomplishment. In the flight test, we conducted using the ducted-fan UAV manufactured in-house. The UAV has the stator which is fixed under main rotor to counteract anti-torque. A ducted-fan UAV is equipped with Pixhawk for flight control and a Raspberry Pi3 for UDP communication.The data communication between the ducted-fan UAVs is performed using Dronekit-python.

1 INTRODUCTION

In recent years UAVs have become popular because they are suitable for many applications such as surveillance and scouting. In addition, with the development of communication technology, multiple UAVs can simultaneously perform missions. This operation has a number of advantages: A group of UAVs can acquire more data and occupy a wider area during surveillance and scouting missions than single UAV [1]. In addtion, the survival rate is higher than when single UAV is operated in the battlefield. Even if rotary UAVs have been widely studied and operated due to its vertical flight capability, they sometimes experience accidents from high-speed rotor blades.

To overcome this risk, the ducted-fan type UAV is introduced, which has a duct to cover the main rotor. Moreover, a well-designed duct can improve thrust efficiency compared with an open-rotor system. This paper proposes to drive a group of ducted-fan UAVs in a formation, using leader-follower strategy for surveillance and scouting [2]. To each UAV the target position is assigned based on the leader UAVs position. The results via ardupilot SITL verifies that the UAVs are able to keep a predetermined formation for mission accomplishment. In the flight test, we conducted using the ducted-fan UAV manufactured in-house [3]. A ducted-fan UAV is equipped with Pixhawk for flight control and a Raspberry Pi3 for UDP communication [4]. The data communication between the ducted-fan UAVs is performed using Dronekit-python.

This paper is organized as : Section 2 describes the formation flight system. Section 3 discusses the formation flight simulation. Section 4 consist of the flight test configuration and flight test results.

2 FORMATION FLIGHT SYSTEM

2.1 Ducted-fan UAV's dynamics and controller design

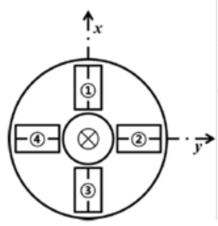
Ducted-fan UAV has an advantage of increasing the thrust efficiency compared to other UAV by equipping a duct. However, unlike the multi-rotor UAV, there is a disadvantage in that it requires to cancel the anti-torque generated by the main rotor. To solve this problem, the ducted-fan UAV compensates for the anti-torque generated by mounting the stator under the rotor [5]. Four vanes located at the end of the duct provide roll, pitch, and yaw control action. The names and effects of each vanes are shown in Figure 1. Equation 1 is ducted-fan UAV's 6 DOF dynamic model equations. Equation 2, 3 shows the ducted-fan UAV's total force and total moment equation.

$$\begin{aligned}
\dot{u} &= vr - wq + (F_x)/m \\
\dot{v} &= wp - ur + (F_y)/m \\
\dot{w} &= uq - vp + (F_z)/m \\
\dot{p} &= \{qr(I_{yy} - I_{zz}) + M_x\}/I_{xx} \\
\dot{q} &= \{pr(I_{zz} - I_{xx}) + M_y\}/I_{yy} \\
\dot{r} &= \{pq(I_{xx} - I_{yy}) + M_z\}/I_{zz}
\end{aligned} \tag{1}$$

where

$$F_{total} = F_{fuse} + F_{prop} + F_{duct} + F_{flap} + F_{grav} \tag{2}$$

$$M_{total} = M_{fuse} + M_{prop} + M_{duct} + M_{flap} + M_{gyro} \tag{3}$$



Sense	Flaps	Effect
$+\delta_{ail}$	①, ③	$+M_x$
$+\delta_{ele}$	②, ④	$+M_y$
$+\delta_{rud}$	①, ②, ③, ④	$+M_z$

Figure 1: control flaps sign conventions.

*Corresponding author : skim78@cnu.ac.kr

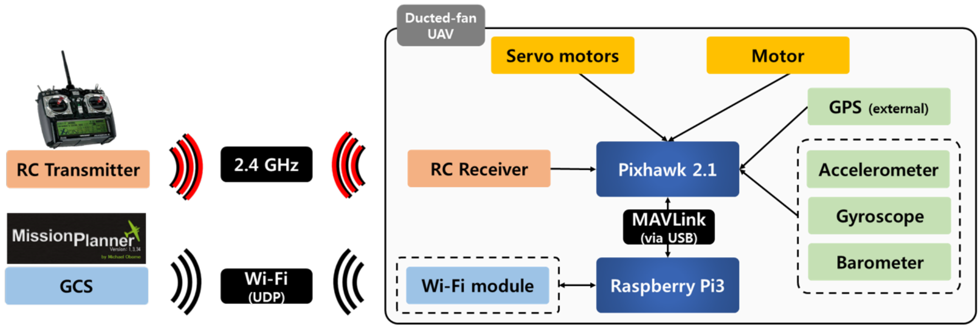


Figure 2: Ducted-fan UAV's components.

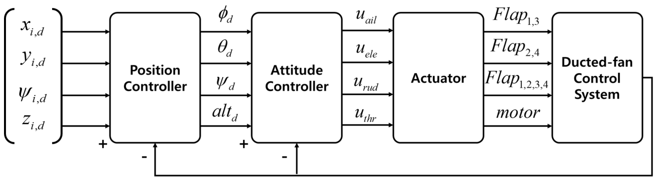


Figure 3: Ducted-fan UAV's control system diagram.

The baseline controller of the ducted-fan UAV is based on PID controller which is provided by arducopter. This attitude controller is designed as an angular position and rate cascaded P-PID inner-outer loop structure. Figure 3 shows the control block diagram of the ducted-fan UAV. The inner loop rate command can be defined as

$$\begin{aligned} e_p &= K_\phi(\phi_d - \phi) - p \\ e_q &= K_\theta(\theta_d - \theta) - q \\ e_r &= K_\psi(\psi_d - \psi) - r \end{aligned} \tag{4}$$

where ϕ_d, θ_d and ψ_d are desired commands, e_p, e_q and e_r are the inner loop rate error. K_ϕ, K_θ and K_ψ are the proportional gains for the Euler angle error for outer loop. Virtual control inputs which act as aileron, elevator and rudder can be calculated as

$$\begin{aligned} u_{ail} &= K_p^p \cdot e_p + K_i^p \int e_p d\tau + K_d^p \frac{de_p}{d\tau} \\ u_{ele} &= K_p^q \cdot e_q + K_i^q \int e_q d\tau + K_d^q \frac{de_q}{d\tau} \\ u_{rud} &= K_p^r \cdot e_r + K_i^r \int e_r d\tau + K_d^r \frac{de_r}{d\tau} \end{aligned} \tag{5}$$

where $K^p_{p,i,d}, K^q_{p,i,d}, K^r_{p,i,d}$ represents proportional, integral and derivative gains of body angular rate (p,q,r) controller. The ducted-fan UAV's position controller is designed as P controller. The position error in the inertial frame can be transformed to the body frame position error depending on the current heading angle (ψ) as follows.

$$\begin{aligned} e_{x\_body} &= e_{x\_in} \cdot \cos\psi + e_{y\_in} \cdot \sin\psi \\ e_{y\_body} &= e_{y\_in} \cdot \cos\psi - e_{x\_in} \cdot \sin\psi \end{aligned} \tag{6}$$

where (e_{x_body}, e_{y_body}) represents the position error in the body coordinate system and (e_{x_in}, e_{y_in}) represents the error in the inertial coordinate system.

2.2 *Development of Ducted-fan UAV*

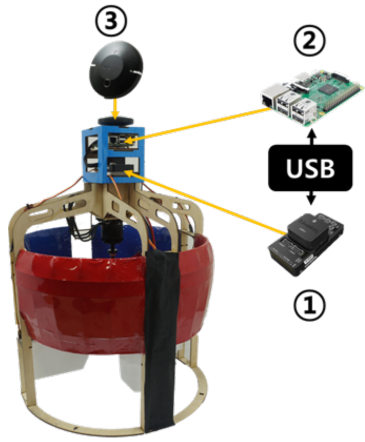


Figure 4: Ducted-fan UAV's equipment.

Figure 4 shows the equipments mounted on the ducted-fan UAV. The ducted-fan UAVs are controlled by an open source flight controller board called ① Pixhawk2.1[6]. The Pixhawk2.1 is equipped with gyroscope, 3D accelerometers and barometer, and it can be connected to ③ an external GPS module. The Pixhawk2.1's software is able to obtain the UAV's attitude and position from these sensors. It enables automatic flight through the obtained attitude data and position data.

In general, when controlling the UAV via Pixhawk2.1, the pilot controls via radio controller or receives commands from a Ground Control System (GCS). However, it is able to communicate with other companion computer via a protocol called MAVLink [7]. In this paper, we use MAVLink by using USB to send commands from a companion computer(② Raspberry Pi3) which is mounted on ducted-fan UAV. In this way, the UAV performs automatic flight as computed by the companion computer. Communication with the GCS is done via UDP communication using the built in Wi-Fi module raspberry pi3. Figure 2 shows the total configuration and architecture connections for the ducted-fan UAV.

2.3 Formation Flight Architecture Design

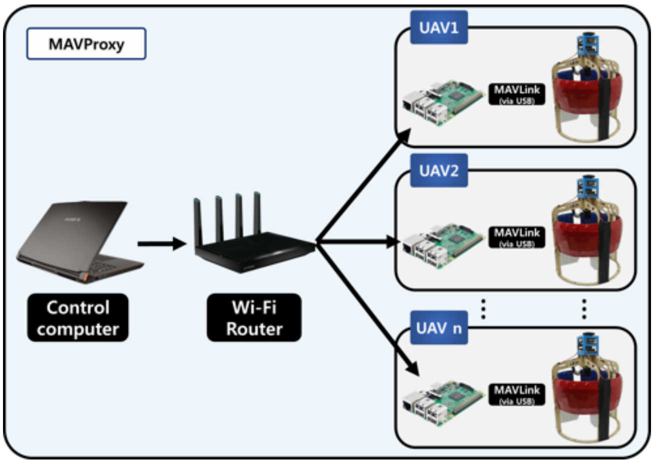


Figure 5: Formation Flight System.

The physical hardware and connections of the flight system are depicted in Figure 5. The system consists of a control computer, a Wi-Fi router, companion computers (Raspberry Pi3) and ducted-fan UAVs. The GCS calculates the formation flight algorithm on the control computer. After that, UDP communication environment is established through Wi-Fi router [8], and the calculated result is transmitted to the raspberry Pi3 mounted on the ducted-fan UAV. The Wi-Fi router connects the control computer to the UAV's companion computers which have different IP and port addresses. Further, UAV companion computers also serve to send those directives to the Pixhawk2.1 via the MAVLink. When the user inputs a formation flight mission on the control computer, the companion computer receives information about the mission and current UAV's flight plan. The flight plan is determined differently according to each UAV, and the corresponding flight plan is transmitted to each Pixhawk2.1. The UAV receives the command and sends information about its current GPS position and attitude status to the companion computer. This UDP communication environment could be constructed through Dronekit-python.

2.4 Dronekit-python framework

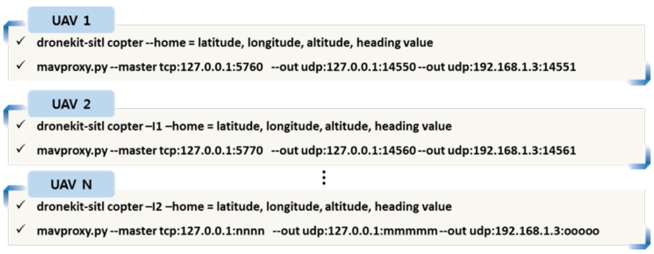


Figure 6: Connect dronekit-sitl and MAVProxy.

Dronekit-python is based on Python which allows developers to create apps that run on a companion computer and communicate with the ArduPilot using a low-latency link [9]. The API communicates with vehicles over 'MAVProxy' which is based on MAVLink [10]. It provides programmatic access to a connected vehicles telemetry and state information, and enables both mission management. In this paper, Dronekit-python runs in the control computer for connecting multi UAVs via UDP communication in order to update the UAV's GPS position data in real time. Figure 6 shows the commands used to connect Dronekit-sitl. It receives data from '– master' and transfers to '– out' IP address.

2.5 'Leader-Follower' formation flight algorithm

The Leader-Follower formation flight is an algorithm that calculates the relative coordinates of Follower UAV based on the position of the Leader UAV and maintains the flight through it. The advantage of this algorithm is that it is easier to construct than other formation flight algorithms because it only needs GPS relative position. Equation 7 represents the relative vector between the Leader UAV positons (X_L) and the Follower UAV positions (X_F) [11]. We constructed the formation to keep the GPS position and barometer altitude data of each UAV in accordance with Equation 1 to describe the relative position for Leader-UAV and Follower-UAV pair at Figure 7.

$$S_{L,F} = X_L - X_F = \begin{bmatrix} latitude_L - latitude_F \\ longitude_L - longitude_F \\ altitude_L - altitude_F \end{bmatrix} \quad (7)$$

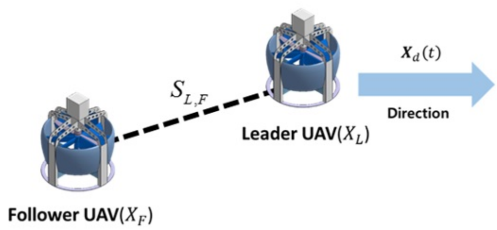


Figure 7: Formation Flight concept drawing.

3 FORMATION FLIGHT SIMULATION

3.1 Simulation Environment

In this part, the Leader-Follower formation flight algorithm written using Dronekit-python is verified via simulations before flight test. The flight simulation model uses the 'singlecopter' model provided by 'Ardupilot' [12]. This model is the same as the firmware used in actual flight. The GCS uses the commercial GCS 'Mission Planner' for UDP communication. The formation flight mission is carried out in the following order.

Figure 8: Formation Flight Simulation GCS Monitor.

- Two UAVs take off to 5m.
- 'Follower UAV' moves to position according to formation flight algorithm.
- It keeps formation flying to the way points.
- After arriving at the end point, 'Leader UAV' do Return to launch(RTL).
- 'Follower UAV' do land at the end point.

In this simulation, the ducted-fan UAV's attitude controller uses a PID controller same as using for flight test. It is assumed that the attitude control of the UAV is well controlled by PID controller's gain tunning. The simulation location was selected as the same filed as the flight test location. The formation interval between the 'Leader UAV' and the 'Follower UAV' follows the relative position vector of Equation 8.

$$S_{L,F} = X_L - X_F = \begin{bmatrix} -0.00007 \\ 0.00007 \\ -2 \end{bmatrix} \tag{8}$$

3.2 Formation Flight Simulation Results

Figure 9 is the formation flight simulation's 2D path graph. $\bullet$ is start point of each UAV, and $\bigcirc$ is destination of each UAV. Follower UAV forms a formation with Leader UAV in order. The Leader UAV returned to launch after it arrived the last point. At this time, the altitude is 15m which is Ardupilot's default setting values. Figure 10 shows the formation flight simulation's 3D path. Simulation results in a slight delay when the 'Follower UAV' follows the 'Leader UAV'. This is due to that the GPS position data of the 'Leader UAV' sent in real time and the 'Follower UAV position' is calculated afterwards. Prior to the ducted-fan UAVs formation flight test, we confirm that the current formation flight algorithm is feasible through this simulation.

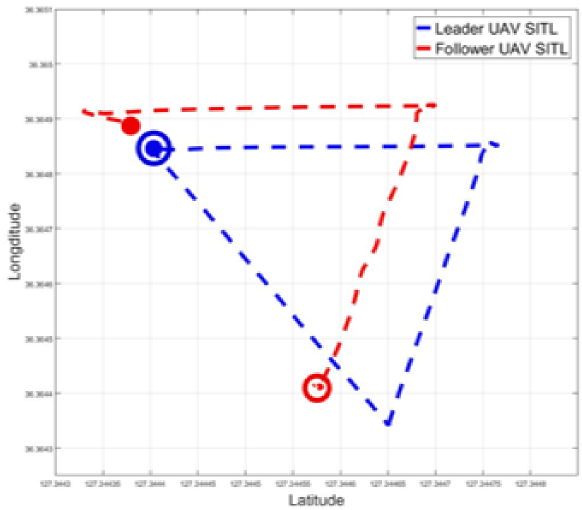


Figure 9: In the simulation 2D path.

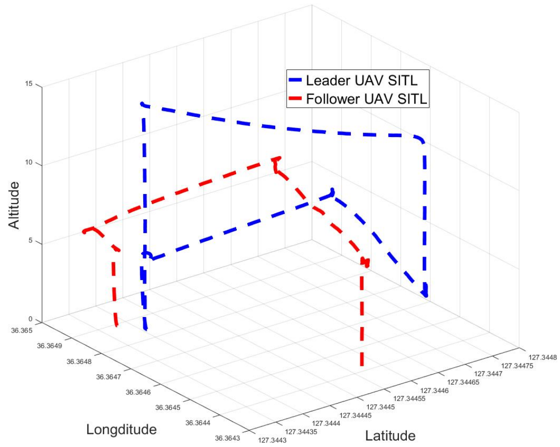


Figure 10: In the simulation 3D path.

4 FORMATION FLIGHT TEST

4.1 Formation Flight Test Setup

This section builds a formation flight system environment for using the formation flight simulation code. Figure 11 shows the configuration required for the formation flight tests. This test performs the same tasks, location and dronekit-python code as the simulation. Two ducted-fan UAVs of the same specification are divided into the leader UAV and the follower UAV. The GCS computer runs the 'Mission Planner' and the Dronekit-python computer runs the dronekit code. The connection between the ducted fan UAV and the two computers uses UDP communication using the MAVProxy in the Wi-Fi environment.

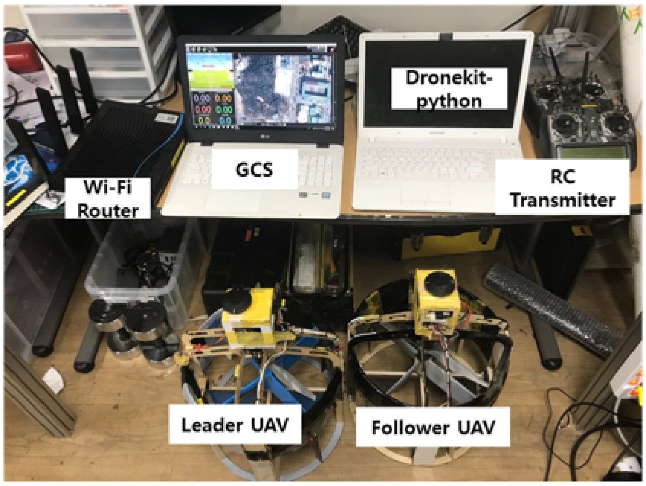


Figure 11: Formation Flight Test Setting.

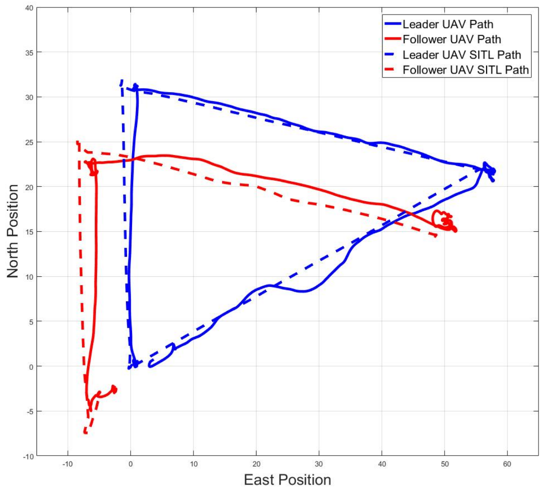


Figure 13: 2D path (Formation Flight Test).

4.2 Formation Flight Test Results

Figure 12: Formation Flight Test 1.

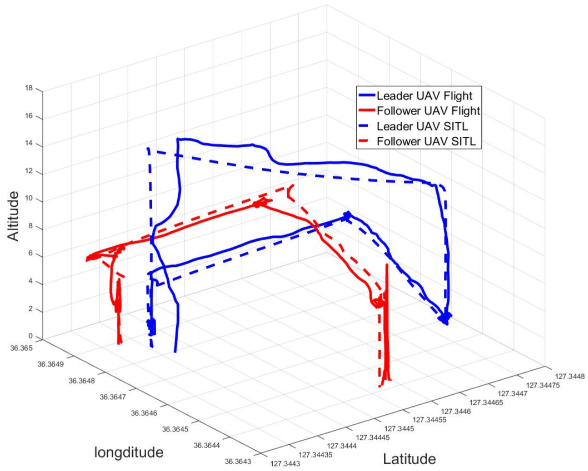


Figure 14: 3D path (Formation Flight Test).

Formation flight test is performed with the same procedure as the simulation. Figure 12 shows the second formation flight path. Figure 13- 14 illustrate 2D and 3D trajectories comparing the second simulation with the actual flight path. The 'Leader UAV' and the 'Follower UAV' followed the commands for keeping their formation. When the 'Leader UAV' returned to launch, it landed at a position away from the target point when landing. This might be caused by GPS sensor error. As a result of the comparison between the two simulations and the flight test, it is confirmed that similar results were obtained when the flight test is performed within the Wi-Fi communication range.

Table 1 shows the mean distance error in the simulation and flight test for the formation flight command. There is an error of about 20%, which is interpreted to be caused by the GPS sensor error and the transmission delay in real time position data.

Table 1: Comparing distance

	CMD	SITL	FlIGHT
Mean distance (m)	10.0995	12.6979	13.8934
Error (%)	–	20.5	27.3

5 Conclusions and Future work

This paper, developed a Leader Follower ducted-fan UAV formation flight algorithm using dronekit-python. In addition, MAVProxy was used to construct formation flight simulation and flight test environment. After that, the simulation was performed and compared with the flight test. In the future, we will supplement the algorithm to reduce the GPS data delay, increase the response rate of the follower, and conduct formation flight tests with number of ducted-fan UAVs.

Acknowledgements

This work is supported by the research project (No. 10062497) funded by the Ministry of Trade, Industry and Energy of Korea, Republic of.

References

[1] Rafael G Braga, Roberto C da Silva, Alexandre CB Ramos, and Felix Mora-Camino. A hybrid approach for 3d formation control in a swarm of uavs using ros.

[2] Zainab Zaheer, Atiya Usmani, Ekram Khan, and Mohammed A Qadeer. Aerial surveillance system using uav. In *Wireless and Optical Communications Networks (WOCN), 2016 Thirteenth International Conference on*, pages 1–7. IEEE, 2016.

[3] Taegyun Kim, Yoonjeong Jang, Minkyu Kim, Junho Jeong, Jinyoung Suk, and Seungkeun Kim. Development and flight test of an electric powered small ducted-fan uav. *KSAS Spring Conference*, pages 210–211, 2017.

[4] Rafael Braga, Alexandre Carlos Brandao Ramos, Roberto Claudino Da Silva, and Félix Mora-Camino. A combined approach for 3d formation control in a multi-uav system using ros. In H. de Plinval J.-M. Moschetta, G. Hattenberger, editor, *International Micro Air Vehicle Conference and Flight Competition 2017*, pages 196–202, Toulouse, France, Sep 2017.

[5] Hong Zhao, Shou-Zhao Sheng, Jian-Bo Li, and Chen-Wu Sun. Modelling and attitude control of a miniature ducted fan uav. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 230(5):953–964, 2016.

[6] Pixhawk homepage. https://pixhawk.org/. Accessed: August, 2018.

[7] Mavlink homepage. https://mavlink.io/en/. Accessed: August, 2018.

[8] Brad Hyeong-Yun Lee, James R Morrison, and Rajnikant Sharma. Multi-uav control testbed for persistent uav presence: Ros gps waypoint tracking package and centralized task allocation capability. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, pages 1742–1750. IEEE, 2017.

[9] Dronekit-python homepage. http://python.dronekit.io/. Accessed: August, 2018.

[10] Mavproxy homepage. https://ardupilot.github.io/MAVProxy/html. Accessed: August, 2018.

[11] Matthew Turpin, Nathan Michael, and Vijay Kumar. Trajectory design and control for aggressive formation flight with quadrotors. *Autonomous Robots*, 33(1-2):143–156, 2012.

[12] Ardupilot homepage. http://ardupilot.org/ardupilot/. Accessed: August, 2018.

Attitude Control Mechanism in an Insect-like Tailless Two-winged Flying Robot by Simultaneous Modulation of Stroke Plane and Wing Twist

H.V. Phan[1], S. Aurecianus[2], T. Kang[2] and H.C Park[1]*
[1]Artificial Muscle Research Center and Department of Smart Vehicle Engineering, Konkuk University, Seoul 05029, South Korea
[2]Department of Aerospace Information Engineering, Konkuk University, Seoul 05029, South Korea

ABSTRACT

In an insect-like tailless flying robot, flapping wings should be able to produce control force as well as flight force to keep the robot staying airborne. This performance requires an active control mechanism that produces sufficient control torques to stabilize the robot due to the inherent instability. In this work, we propose a control mechanism integrated in a hovering-capable, two-winged, flapping-wing, 17.6 g flying robot (KUBeetle-S) that can simultaneously change the wing stroke-plane and wing twist. The mechanism is capable of tilting the stroke plane causing change in the wing twist to produce coupling control torques for pitch and roll. For yaw (heading change), the root spars of the left and right wings are adjusted asymmetrically to change the wing twist during flapping motion, resulting in yaw torque generation. We first performed a series of experiments using a 6-axis force/torque load cell to evaluate the effectiveness of the control mechanism via torque generation. We then prototyped the robot integrating the control mechanism with sub-micro servos as actuators and flight control board, and conducted free flight tests to verify the possibility of attitude control.

1 INTRODUCTION

In the absence of tail control surface, insects are capable of modifying their wing kinematics to produce control forces for attitude change during flight [1,2]. In particular, shifting the stroke plane position or flapping angle ranges for pitch response was found in fruit fly *Drosophila melanogaster* [3] and hoverflies *Eristalis tenax* and *Episyrphus balteatus* [4]. For roll and yaw responses, many insect species such as *Drosophila spp*. [5], *Musca domestica* [6], *Calliphora spp.* [7], dragonflies [8], beetles [9], and hawkmoths [10], are found to change their flapping amplitudes of two wings.

Mimicking those complex kinematics manipulation abilities is a challenging task in developing light-weight insect-like tailless flapping-wing flying robots. Without tail stabilizer, the main flapping wings should be incorporated with a control mechanism to produce control forces as well as sufficient flight force to keep the robot staying airborne. Within a limited takeoff weight of the robot, a proposed control mechanism should be fitted for light-weight actuators, which generate a low actuation torque. Additionally, the mechanism is also required to produce enough control torques for attitude changes. Due to the hurdles, there are only a few hovering tailless flying robots ready for free flight although many research groups have successfully developed bird-like tailed robots [11-15]. A 19 g Nano Hummingbird developed by AeroVironment

*contact email: hcpark@konkuk.ac.kr

[16] is the first tailless two-winged robot that successfully performs stable controlled flight. It utilizes a wing twist modulation mechanism by controlling the wing root spars symmetrically or asymmetrically for roll, pitch, and yaw controls. Similar wing twist modulation mechanism can be found in a 21 g KUBeetle [17] and a 22 g Colibri robot [18]. Otherwise, wing kinematics modulation was used as a control approach in a tiny 80 mg Robobee [19] and a 12 g robotic hummingbird [20]. In this approach, wing stroke amplitudes of the left and right wings are changed asymmetrically for roll control. By shifting the mean stroke angle of the two wings forward or aft, pitch control is obtained. In addition, modulation of stroke velocities in each half-stroke within a flapping cycle results in a yaw torque generation for yaw control. Another flight control approach is the wing stroke-plane modulation for pitch and yaw controls used in a 62 g Robotic Hummingbird [21].

We have been also developing a tailless flying robot KUBeetle, which performed stable flight for about 40 s [17]. We have tested for many control approaches including modulations of mean stroke angle [22], stroke-plane angle [23], or wing twist [17]. However, these control mechanisms were complicated for fabrication and required high-torque actuators. In this work, we propose a simple and lightweight control mechanism that is able to modify synchronously the stroke plane and wing twist for attitude changes. We design and fabricate the mechanism, and evaluate its possibility of control force and torque generation using a 6-axis force/torque load cell. Then, the control mechanism, which is actuated by three micro servos, is integrated in the flying robot for free controlled flight test.

2 FLAPPING-WING MECHANISM

The tailless robot is developed aiming to mimic the flight of a horned beetle, *Allomyrina Dichotoma,* who is capable of hovering flight. Since the mechanical design of the flapping-wing mechanism was presented in detail in [17], this paper shows only a brief summary. Two deformable wings are actuated by a 3.5 g coreless motor (Chaoli CL720, China) through a gearbox (24:1) to amplify the torque of the motor and a transmission linkage system, which converts the rotary motion to the flapping motions using a combined operation of the 4-bar linkage and the pulley-string mechanism. The wing with a length of 75 mm was made of 10 μm Mylar film as wing membranes and one-layer carbon strips as reinforced veins. After installing to the flapping mechanism, the two wings are able to deform creating spanwise twist and chordwise camber during the flapping motion [16]. Flapping amplitude is approximately 190° allowing the presence of clap-and-fling effect for vertical force augmentation [24]. Effect of vein structures on force generation and power requirement was also investigated [25].

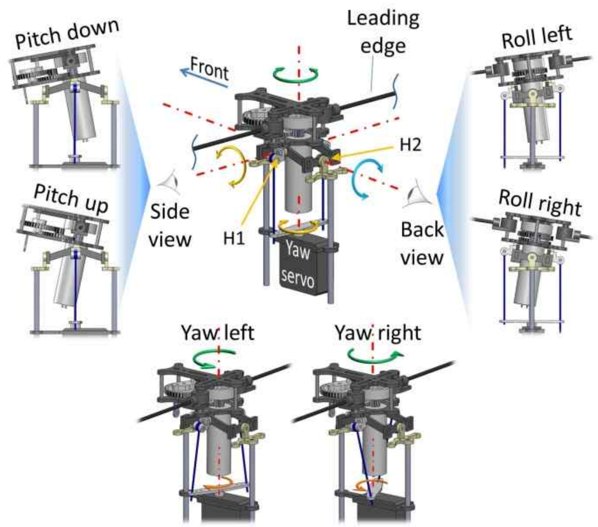


Figure 1 – Conceptual design of the control mechanism for attitude change.

3 ATTITUDE CONTROL MECHANISM

3.1 Conceptual design and fabrication

Figure 1 shows CAD images of the flight control mechanisms for pitch, roll and yaw attitude changes. The flapping-wing mechanism (including the motor) is able to rotate about two hinges (H1 and H2 in Figure 1) for pitch and roll controls. By tilting the flapping-wing mechanism around the hinge H1 forward or backward, the wing stroke plane is tilted in the same direction to the change in the directions of forces for pitch torque generation. Additionally, since the wing root spars are fixed, in this case, tilting the stroke plane causes the modulation in wing twist to produce additional pitch torque. To generate roll control torque, the flapping-wing mechanism is rotated about hinge H2 to tilt the stroke plane laterally to the left or right (Figure 1). As a result, wing twists of the two wings are also modulated asymmetrically. Thus, pitch and roll control torques are generated by simultaneously changing the stroke plane and modulating the wing twist. Yaw motion is controlled by rotating the root spars of the left and right wings in the opposite directions, as shown in Figure 1.

Based on the conceptual design, we fabricated the control mechanism using a 0.8 mm carbon/epoxy panel and installed in the flying robot, as depicted in Figure 2. Pitch and yaw controls are actuated independently by two conventional ultra-micro digital servos (HK-5320, hobbyking.com) weighing 1.5 g. Meanwhile, roll control is actuated by a sub-micro LZ servo (0.5 g, microflierradio.com).

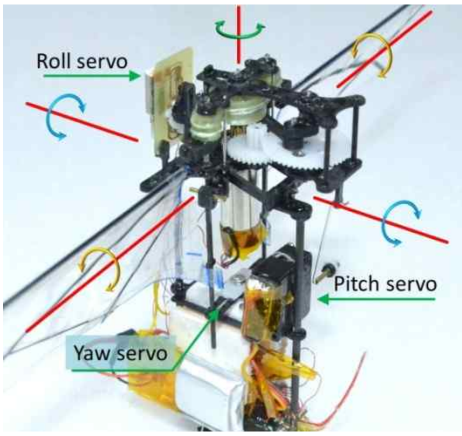


Figure 2 - Prototype of the control mechanism integrated in the flying robot.

3.2 Force and torque generation

To investigate the capability of control torques generation, we set up an experiment using a 6-axis load cell (Nano 17, ATI Industrial Automation, USA, force resolution ≈ 0.3 gf, torque resolution ≈ 0.0156 N.mm), as shown in Figure 3. We located the load cell close to the location of the robot's center of gravity (CG). Thus, the torques obtained from the load cell can be regarded as those about the CG. We powered the flapping-wing system using an external power supply (E36103A, Keysight, Korea) at a flapping frequency of 23 Hz.

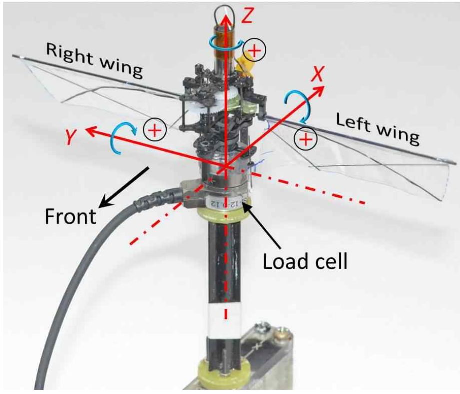


Figure 3 – Experimental setup for forces and torques measurement using a load cell

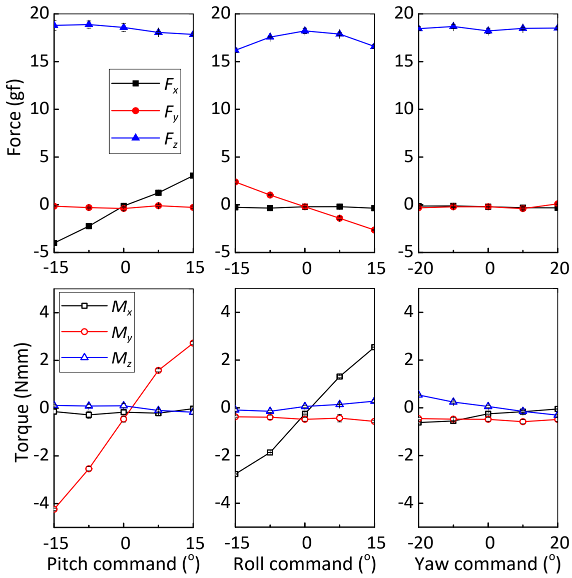


Figure 4 – Generation of forces and torques in three axes for a range of control inputs.

Figure 4 shows the forces and torques generated in all three axes for different pitch, roll and yaw inputs. At 23 Hz, a vertical force of about 18.5 gf was produced. Changing control inputs for pitch resulted in linear changes in horizontal force and pitch torque. In the range of control input (stroke-plane angle) from –15° (pitch down) to 15° (pitch up), the measured horizontal force varied from –4.0 gf to 3.1 gf and pitch torque varied from –4.2

N.mm to 2.7 N.mm. With no control inputs for hovering condition, pitch torque of about -0.5 Nmm was generated. This is due to slight vertical misalignment between the mean aerodynamic force center and the load cell. Effect of pitch control inputs on other force and torque signals was insignificant. For roll control, changing the control input from –15° (roll right) to 15° (roll left) resulted in the changes of lateral force from 2.4 gf to –2.6 gf, and roll torque from –2.8 N.mm to 2.6 N.mm. A slight decrease in the vertical force was found for a higher roll commend angle. Yaw torque is less sensitive to the yaw command compared to those of pitch and roll commands. For the range of yaw inputs by the tilt of yaw servo's arm (Figure 1) from -20° to 20°, yaw torque varied from 0.5 N.mm to -0.3 N.mm (a reference of 0.06 N.mm at hovering condition).

4 FLIGHT EXPERIMENT

4.1 Attitude stabilization

For attitude control and stabilization, a custom-built 1 g control board [26] was mounted onboard. The board consists of a microprocessor ARM Cortex-M4 32-bit STM32L432KC, a 6-axis gyroscope and accelerometer MPU-9250, a 2.4GHz transceiver nRF24L01+, and power regulators, as shown in Figure 5. Due to the inherent instability of the tailless robot, a feedback PD controller was implemented in the control system to sense the attitudes of the robot, which can be determined by the roll, pitch, yaw Euler angles (ϕ, ϑ, ψ, respectively), and angular rates (p, q, r, respectively) about the *X*-, *Y*-, and *Z*-axis, respectively. Accelerometer readings are used to obtain the roll and pitch angles. Meanwhile, the angular rates are estimated from the gyroscope readings. Thus, the roll and pitch attitudes can be estimated either by an accelerometer or a gyroscope. However, the data obtained from the accelerometers are strongly affected by vibrations caused by a high flapping motion. The gyroscope signals, on the other hand, are less sensitive to disturbances, but drift by time. To solve this issue, a combination of low-pass and Kalman filters, which uses the signals from both gyroscope and accelerometer, was used to filter the roll and pitch angles, while a low-pass filter was used to smooth the yaw signal [26].

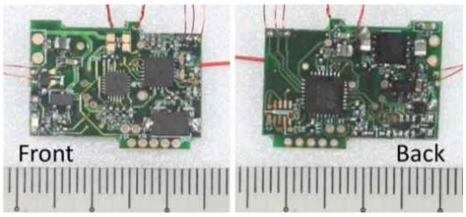


Figure 5 – Custom-built flight control board used in the flying robot.

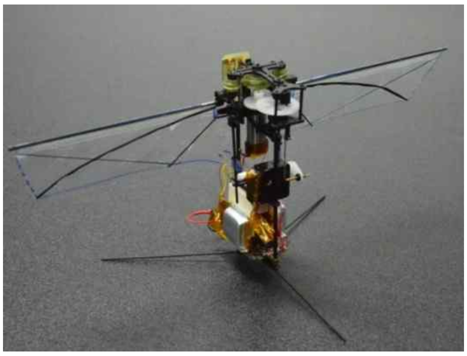


Figure 6 – Overview of the KUBeetle-S integrated with onboard control system.

Component	Weight (g)	Percentage (%)
Flapping mechanism	3.0	17.0
Driving motor	3.5	19.9
Control servos	3.8	21.6
Control mechanism	0.5	2.8
Batteries	4.2	23.9
Control board	1.0	5.7
Supporting frames	0.9	5.1
Wings	0.4	2.3
Wires	0.3	1.7
Total mass	**17.6**	

Table 1 - Mass breakdown of the KUBeetle-S

4.2 Characteristics of the flying robot

Table 1 shows the weight breakdown of the robot, which is named KUBeetle-S, used for flight. The robot with a wingspan of 170 mm and height of 75 mm (Figure 6) weighs about 17.6 g. The driving motor was powered by a two-cell lithium-polymer battery connected in series (7.4 V, 70 mAh). With onboard regulators in the control board, the servos and MPU9250 sensor were supplied by 5V and 3.7V sources, respectively. The power inputted to the motor was manually controlled though the power throttle stick.

4.3 Flight tests

The KUBeetle-S was tested for its free flight to evaluate the effectiveness of the control mechanism. We recorded the flight using three synchronized high-speed cameras (Photron Ultima APX, 1024 x 1024 pixels, 250 fps). The flight trajectory and body attitude (roll, pitch, and yaw angles) of the robot were obtained using DLT program [27]. Due to the limitation in the recording time of the high-speed cameras, the flight used for the analysis was lasted in 8 seconds. Figure 7 shows the composite images and trajectory of the robot during its flight. Its body attitude is shown in Figure 8. Pitch and roll angles showed small variations from the reference of 0°. Yaw motion was stabilized using gyroscope signal only, which experience drift during flight. Additionally, yaw control torque is less sensitive to the yaw input, as shown in Figure 4. Therefore, heading direction is drifted during flight. However, the upright stability of the robot was unaffected by the heading stability. With the battery capacity, the robot demonstrated its successful flight for more than 2.5 minutes. The flight thus proved that the attitude control system is successfully implemented in an insect-like tailless flying robot.

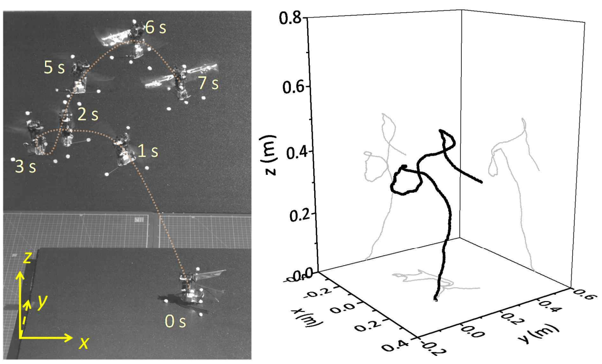


Figure 7 –Composite images and three-dimensional trajectory of the KUBeetle-S during hovering flight.

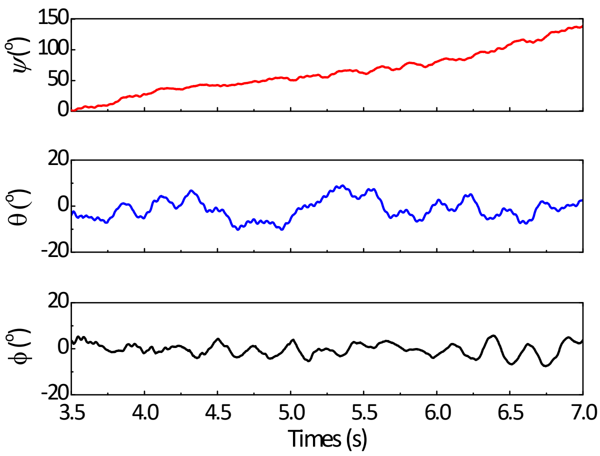


Figure 8 – Attitude performance of the KUBeetle-S during hovering flight.

5 CONCLUSION

This work introduced a 17.6 g insect-like tailless, two-winged, hovering-capable KUBeetle-S robot that can change its stroke plane and wing twist simultaneously for pitch and roll controls, and modulate the wing root spars asymmetrically for yaw control. The proposed control mechanism, which requires less actuation torques from actuators, is simple and easy for fabrication allowing us to use small actuators for saving weight. The measured force and torque proved that the control mechanism can effectively generate reasonable amount of force and torque for attitude changes. Finally, the KUBeetle-S with implementation of onboard attitude feedback control system successfully hovered and loitered for more than 2.5 mins, demonstrating the effectiveness of the proposed control mechanism.

ACKNOWLEDGEMENTS

This research was partially supported by the 2018 KU Brain Pool Program of Konkuk University, Korea.

REFERENCES

[1] R. Dudley. The biomechanics of insect flight: Form, function, evolution. Princeton, NJ: Princeton University Press, 2002.

[2] G. K. Taylor. Mechanics and aerodynamics of insect flight control. Biological Reviews, 76:449-471, 2001.

[3] J. M. Zanker. On the mechanism of speed and altitude control in *Drosophila melanogaster*. Physiological Entomology, 13:351–361, 1988.

[4] C. P. Ellington. The aerodynamics of hovering insect flight. I. The quasi-steady analysis. Philosophical Transactions of the Royal Society B, 305:1-15, 1984.

[5] S. Vogel. Flight in *Drosophila*. II. Variations in stroke parameters and wing contour. Journal of Experimental Biology, 46:383–392, 1967.

[6] M. V. Srinivasan. A visually-evoked roll response in the housefly. Open-loop and closed-loop studies. Journal of Comparative Physiology A, 119:1–14, 1977.

[7] W. Nachtigall, and W. Roth. Correlations between stationary measurable parameters of wing movement and aerodynamic force production in the blowfly (*Calliphora vicina* R.-D.). Journal of Comparative Physiology A, 150:251–260, 1983.

[8] D. E. Alexander. Unusual phase relationships between the forewings and hindwings in flying dragonflies. Journal of Experimental Biology, 109:379–383, 1984.

[9] A. J. Burton. Directional change in a flying beetle. Journal of Experimental Biology, 54:575–585, 1971.

[10] A. E. Kammer. The motor output during turning flight in a hawkmoth *Manduca sexta*. Journal of Insect Physiology, 17:1073–1086, 1971.

[11] J. H. Park, and K. J. Yoon. Designing a biomimetic ornithopter capable of sustained and controlled flight. Journal of Bionic Engineering, 5 (1):39-47, 2008.

[12] G. C. H. E. de Croon, M. A. Groen, C. D. Wagter, B. Remes, R. Ruijsink, and B. W. van Oudheusden. Design, aerodynamics and autonomy of the DelFly. Bioinspiration & Biomimetics, 7:025003, 2012.

[13] Festo, SmartBird - Aerodynamic lightweight design with active torsion, https://www.festo.com/net/SupportPortal/Files/46270/Brosch_SmartBird_en_8s_RZ_110311_lo.pdf, 2011.

[14] S. S. Baek, F. L. G. Bermudez, and R. S. Fearing. Flight control for target seeking by 13 gram ornithopter. In IEEE Conference on Intelligent Robots and Systems (IROS), 2011.

[15] Q. V. Nguyen, W. L. Chan, and M. Debiasi. Hybrid design and performance tests of a hovering insect-inspired flapping-wing micro aerial vehicle. Journal of Bionic Engineering, 13(4):235-248, 2016.

[16] M. T. 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.

[17] H.V. Phan, T.S. Kang, and H.C. Park. Design and stable flight of a 21 g insect-like tailless flapping wing micro air vehicle with angular rates feedback control. Bioinspiration & Biomimetics, 12:036006, 2017.

[18] A. Roshanbin, H. Altartouri, M. Karásek and A. Preumont. Colibri: A hovering flapping twin-wing robot. International Journal of Micro Air Vehicles, 9(4):270-282, 2017.

[19] 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.

[20] J. Zhang, Z. Tu, F. Fei, and X. Deng. Geometric flight control of a hovering robotic hummingbird. In IEEE Conference on Robotics and Automation (ICRA), 2017.

[21] D. Coleman, M. Benedict, and I. Chopra. Design, Development and Flight Testing of Robotic Hummingbird. In 71st Annual Forum of the American Helicopter Society, 2015.

[22] Q. T. Truong, H. V. 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, 2016.

[23] H. V. Phan, and H. C. Park. Generation of Control Moments in an Insect-like Tailless Flapping-wing Micro Air Vehicle by Changing the Stroke-plane Angle. Journal of Bionic Engineering, 13(3):449-457, 2016.

[24] H. V. Phan, T. K. L. Au, H. C. Park. Clap-and-fling mechanism in a hovering insect-like two-winged flapping-wing micro air vehicle. Royal Society Open Science, 3:160746, 2016.

[25] H. V. Phan, and H. C. Park. Design and evaluation of a deformable wing configuration for economical hovering flight of an insect-like tailless flying robot. Bioinspiration & Biomimetics, 13:036009, 2018.

[26] S. Aurecianus, H. V. Phan, N. S. Lok, T. Kang, and H. C. Park. Flapping wing micro air vehicle (FW-MAV) state estimation and control with heading and altitude hold. In 10th International Micro Air Vehicle Competition and Conference (IMAV), 2018.

[27] T. Hedrick. Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems. Bioinspiration & Biomimetics, 3:034001, 2008.

Aerodynamics during forward flight of a tailless flapping-wing micro air vehicle

L.T.K. Au and H.C. Park*
Konkuk University, Department of Smart Vehicle Engineering

ABSTRACT

In this paper, the aerodynamic performance during forward flight of KU-Beetle—an insect-like flapping-wing micro air vehicle (FW-MAV) is studied. A range of advance ratio (*J*) from 0 to 0.5 was considered. The aerodynamic forces and pitching moment could be manipulated by adjusting two parameters: the wing-root angle (γ) and stroke plane angle (β). For each investigated advance ratio, the aerodynamic forces and pitching moment for a pair of β and γ was computed by computational fluid dynamics (CFD) method via commercial software of ANSYS Fluent 16.2. The average values taken at third flapping cycle when the flow was settled could be obtained. The equilibrium for an advance ratio, in which the drag and pitching moment on the body were balanced by horizontal force and pitching moment produced by the wing respectively, was acquired using Newton-Raphson method. The study shows that for flapping angle (ψ) in range from -90° to 90°, the inflow due to forward flight augmented the total inflow velocity during downstroke, and reduced the total inflow during upstroke. This results in "reverse region"—a part of the wing starting from wing root where the total inflow direction during upstroke was reversed. In this reverse region, the lift was negative, which reduced the lift produced by the whole wings. The region enlarged when the forward flight speed increased. Therefore, compared to hovering, for the same stroke plane angle, during forward flight, the wings produced more drag during upstroke and less thrust during downstroke, resulting in larger drag; meanwhile, the wings produced more lift during upstroke and less lift during downstroke, hence the change of lift is insignificant. To balance drag during forward flight, KU-Beetle must incline forward, so that the horizontal component of the lift can overcome the drag. When *J*=0.5, the body inclined 40° which is a little larger than that of a bumble bee.

1 INTRODUCTION

Insects' extraordinary flight ability has drawn attention of scientists for over a century. It has been proven that the conventional aerodynamics model based on translational force cannot estimate sufficient lift for hovering insects [1]. Therefore, numerous efforts have been carried out to reveal the underlying force augmentation mechanisms of flapping flight, and considerable progress in aerodynamics of insect flight has been achieved in recent decades. These include clap-and-fling, leading edge vortex created by delayed stall, rotational circulation and wake capture functioning during stroke reversal and added mass. The influences of wing-wake interaction, wing-wing interaction on the production of aerodynamic forces in four-winged insects were also widely studied. With these discoveries, quasi-steady aerodynamic model was modified to improve the accuracy in estimation of aerodynamic force produced by flapping wings [2].

Among the flight regimes of insect flight, hovering received the most attention from scientists. This flight regime is the first goal in most flapping-wing micro air vehicle development. There are also considerable studies in forward flight, in both areas of aerodynamics and dynamic flight stability. Recently, Han et al. proposed a semi-empirical quasi-steady aerodynamic model for

*contact email: hcpark@konkuk.ac.kr

force estimation during forward flight of flapping flyers [3]. Although forward flight in insects has been extensively studied, the aerodynamics during forward flight of a flapping-wing micro air vehicle (FW-MAV) remains limited in literature survey. In this paper, we report the aerodynamic performance in forward flight of KU-Beetle, a tailless FW-MAV. A range of advance ratios (J), which is the ratio between the forward flight speed and the mean tip speed, from 0 to 0.5 was investigated.

2 MATERIAL AND METHOD

Section headings should be numbered, centre justified and in all capitals. Font size shall be 11pt. Sub-headings are left justified, numbered and italic.

2.1 Wing kinematics

In KU-Beetle, the aerodynamic forces and pitching moment can be manipulated by changing the angle γ between the wing root and the body line, as shown in Fig. 1 (Phan et al. 2016 [4]). When γ is changed, the wing deformation is adjusted, resulting in control forces and pitching moment.

The motion of the left wing is described in Figure 2a. The wing simultaneously flaps around the flapping axis (z-axis) and pitches around the feather axis (ξ-axis). The feather axis is attached to the leading edge of the wing. The location of the feather axis is determined by the flapping angle ψ. Each wing section was treated as a parabolic curve whose shape could be determined by two parameters. The first one is the mid-chord rotation angle ϑ_m—the angle between the stroke plane and the line connecting the leading edge to the mid-chord position. The second one is the full-chord rotation angle ϑ_r—the angle between the stroke plane and the line connecting the leading edge to the trailing edge. The variations of ψ, ϑ_m and ϑ_r versus time were captured by hi-speed cameras. These data were fitted by sums of sinusoidal functions.

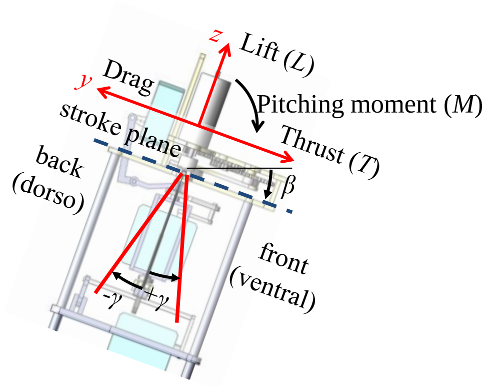


Figure 1 - Control variable, the red lines indicate the wing root

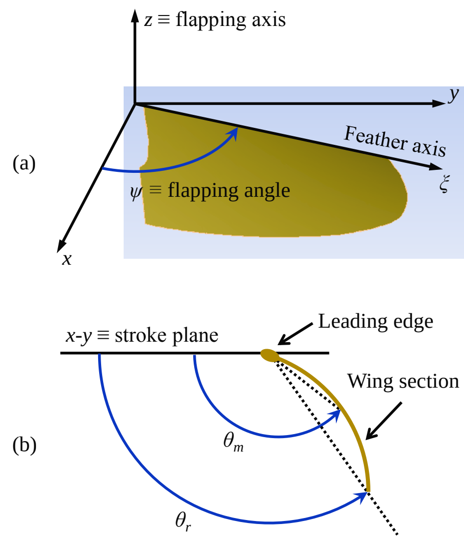


Figure 2 - Wing motion. (a) Three-dimensional wing motion. (b) Wing camber

2.2 Computational fluid dynamics model

The forces and pitching moments acting on the wing and the body frame were computed separately using CFD method. Figure 3 illustrates the computational domain and wing geometry. The wing was modeled as a membrane twisted from the root to tip. Because of longitudinal symmetrical plane, the CFD model was built for only left wing. Therefore, the computational domain is a half cylinder. The size of the computational domain was chosen such that the diameter and the length are twelve times the distance from the wing root to tip (wing length).

The mesh is finest around the wing, and become coarser toward the far-filed region, as shown in

Fig. 4. A high density region whose diameter doubles the wing length was built around the wing. The wing surface was meshed by approximately 21,000 triangle elements.

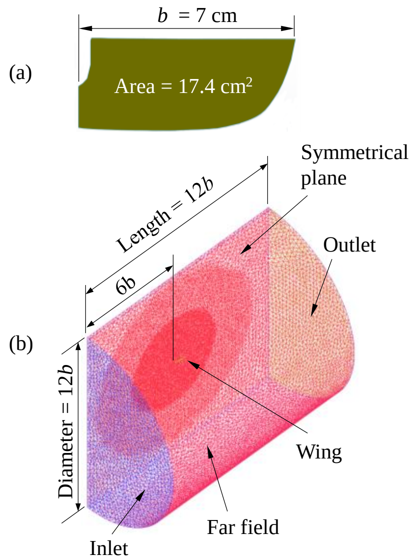


Figure 3 - (a) Wing geometry. (b) Computational domain

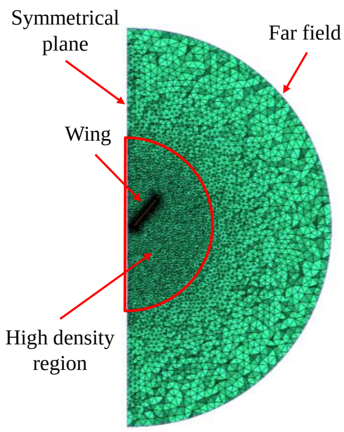


Figure 4 - A cross section of the volume mesh

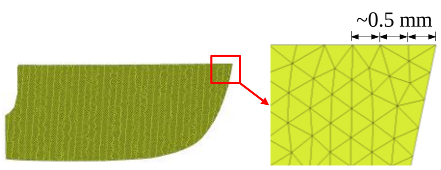


Figure 5 - Mesh on the wing surface

3 RESULT AND DISCUSSION

3.1 Effect of forward flight on force generation

The total inflow velocity is expressed as follows:

$$\vec{V}_{inflow} = V_{inflow}\vec{e}_T = V_T\vec{e}_T + U\vec{e}_T, \tag{1}$$

where V_T and U are the inflows due to flapping motion and due to forward flight, respectively, and $\vec{e}_t$ is the unit vector points in direction of inflow due to translational motion. These quantities can be expressed as follows:

$$V_T = \begin{cases} -r\dot{\psi} \text{ during downstroke} \\ r\dot{\psi} \text{ during upstroke} \end{cases}$$
$$U = \begin{cases} V_f \cos\psi \text{ during downstroke} \\ -V_f \cos\psi \text{ during upstroke} \end{cases} \tag{1}$$

Therefore, V_{inflow} can be expressed as follows:

$$V_{inflow} = \begin{cases} -r\dot{\psi} + V_f \cos\psi \text{ during downstroke} \\ r\dot{\psi} - V_f \cos\psi \text{ during upstroke} \end{cases} \tag{2}$$

A negative V_{inflow} means that the inflow is reversed, and the translation lift produced by the wing is negative. The total inflow during downstroke and upstroke are illustrated in Fig. 6. For -90<ψ<90°, V_f $cos\psi$>0, hence, the inflow due to forward flight augments the total inflow velocity during downstroke, and reduces the velocity during upstroke, as shown in Fig. 6a,b. On the other hand, when ψ<-90° or ψ>90°, V_f $cos\psi$>0, hence, the inflow due to forward flight reduces the total inflow velocity during downstroke, and augments the velocity during upstroke, as shown in Fig. 6c,d. As a result, for -90<ψ<90°, the reverse region appears during upstroke, while for ψ<-90° or ψ>90°, the region appears during downstroke. The flapping angle of current FW-MAV is from -93° to 90°. That means the portion when ψ<-90° or ψ>90° is negligible compared to that when -90<ψ<90°. As a result, in most of the flapping cycle, the reversed flow appears during upstroke, which explains the reduction in lift and thrust during upstroke, and augmentation in lift and drag during downstroke, as plotted in Fig. 7.

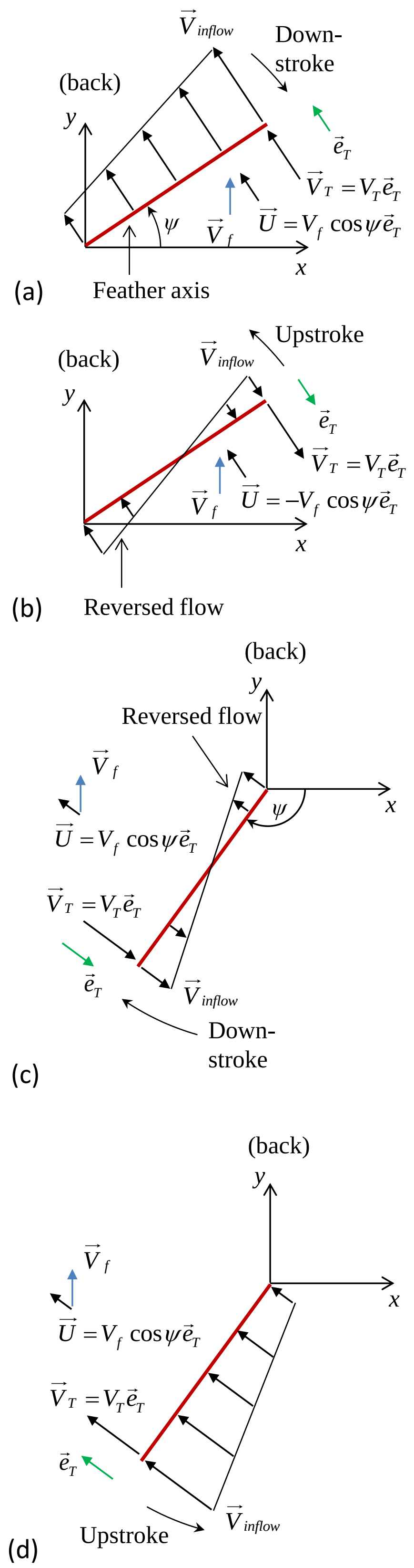


Figure 6 - Inflow in KU-Beetle. (a) During downstroke, −90<ψ<90°. (b) During upstroke, −90<ψ<90°. (c) During downstroke, ψ<−90° or ψ>90°. (d) During upstroke, ψ<−90° or ψ>90°.

The mean lifts and drags for various advance ratios *J* ranging from 0 to 0.5 are listed in table 1. As *J* increases, the mean lift and drag increase during downstroke, while the lift and thrust decreases during upstroke. As a result, the total mean lift over one flapping cycle does not change much. while the total mean drag increases when *J* enlarges.

	Stroke	*J*=0	*J*=0.25	*J*=0.5
Lift [gf]	Down	9.23	16.63	23.22
	Up	10.46	7.00	5.90
Drag [gf]	Stroke	J=0	J=0.25	J=0.5
	Down	5.20	10.93	16.97
	Up	-5.56	-2.88	-2.42

Table 1 - Mean lift and drag for various *J*s

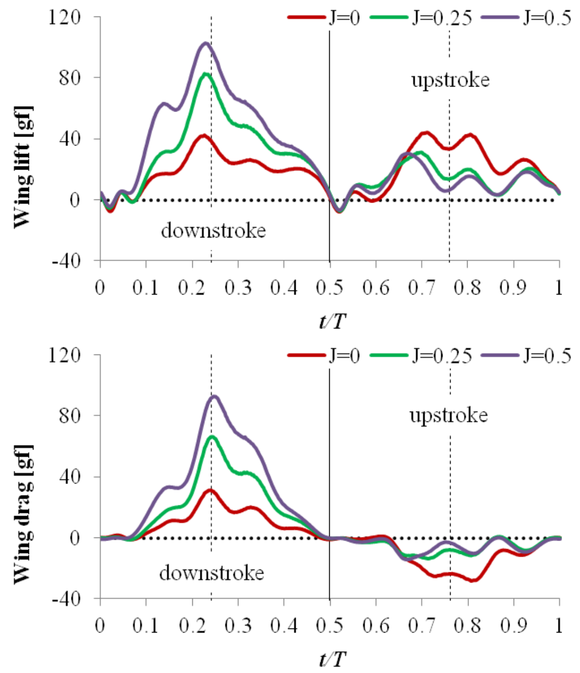


Figure 7 - Time-course lift and drag for various *J*s

3.2 Equilibrium forward flight

For each advance ratio *J*, the equilibrium flight condition can be achieved by adjusting the wing root angle γ and the stroke plane angle β. The pair (γ,β) can be found using Newton-Raphson method. Let *H* be the force in backward horizontal direction, and *M* be the pitching moment. At equilibrium, the total *H* and *M* of the body and the wings equal to 0. The flow chart using Newton-Raphson algorithm for acquiring the pair (γ_e,β_e)–values of γ and β when the equilibrium for an advance ratio *J* is achieved–is illustrated in Fig. 8.

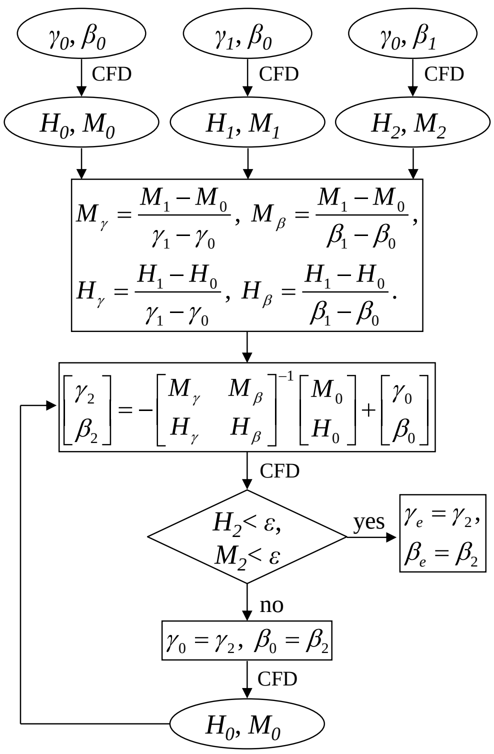


Figure 8 - Flow chart for acquiring equilibrium

The equilibrium stroke plane angle for various *J* is plotted in Fig. 9. The result of bumble bee is plotted in the same figure for comparison. For forward flight, KU-Beetle must incline forward, so that the horizontal component of the lift can overcome the increasing drag on the wing and the body frame. This result has similar trend to that of bumble bee.

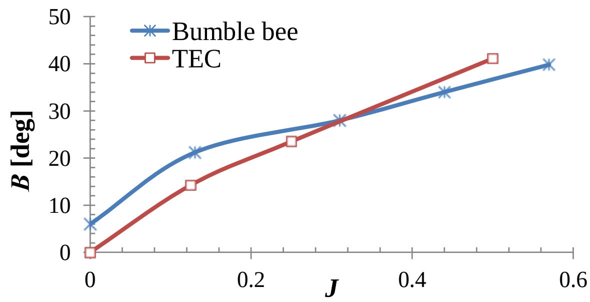


Figure 9 - Stroke plane angle for equilibrium during forward flight

4 CONCLUSION

The aerodynamics during forward flight of KU-Beetle—a tailless FW-MAV was investigated using CFD method. Because of the inflow due to forward flight, the inflow increases during downstroke, and decreases during upstroke. This results in augmentation in lift and drag during downstroke, and reduction in lift and thrust during upstroke. The equilibrium flight condition was acquired using Newton-Raphson method. To achieve equilibrium as the forward flight speed increases, KU-Beetle must incline forward, which means that the stroke plane angle increases.

ACKNOWLEDGEMENTS

This research was partially supported by the KU Brain Pool Program of Konkuk University.

REFERENCES

[1] Ellington C P. The aerodynamics of hovering insect flight. I. I. The quasi-steady analysis. *Philosophical Transactions of the Royal Society of London B*, 1984, **305**, 1-15

[2] Truong Q T, Nguyen Q V, Truong V T, Park H C, Byun D, Goo N S. A modified blade element theory for estimation of forces generated by a beetle-mimicking flapping wing system. *Bioinspiration & Biomimetics*, 2011, **6**, 1-11.

[3] Han J S, Chang J W, Han J H. An aerodynamic model for insect flapping wings in forward

flight. *Bioinspiration & Biomimetics*,2017, **12**, 1-16.

[4] H. P. Phan, T. Kang and H. C. Park, "Design and stable flight of a 21 g insect-like tailless flapping wing micro air vehicle with angular rates feedback control," *Bioinspiration & Biomimetics*, volume 12, 2017, 17pp. Dynamic flight stability of a hovering bumblebee. *Journal of Experimental Biology*, 2005, **208**, 447-459.

Development of a 200 mg bio-inspired nano-flying robot

T. Jardin,* S. Prothin, P. Barricau and J.R. Frutos
ISAE-SUPAERO, Université de Toulouse, France
SilMach, Besançon, France

ABSTRACT

We report on the development of a 200 mg flapping wing robot. The flapping motion is driven by a simple rod-crank system connected to an electrostatic motor operating at frequencies up to a few Hertz. The wings are connected to the rod-crank system *via* a few tens of μm thick hinge which allows passive wing rotation at stroke reversal. Components of the robot are manufactured out of silicon, Titane, PEEK and Parylene to reduce weight and friction. A test bench is developped that allows to reproduce the wing dynamics of the 200 mg prototype, with enhanced robustness and larger range of operating frequency. Three systems with 60, 90 and 120° flapping amplitudes and interchangeable wings operating at frequencies up to 30 Hertz are used for long lasting experiments. The dynamics of the wings are analyzed using High Speed Camera Visualization (HSV) and aerodynamic loads are derived from Particle Image Velocimetry (PIV) measurements in the wake of the robot and compared with balance measurements. Kinematics of the wings obtained from HSV are then implemented to extract forces and 3D velocity flow fields from Direct Numerical Simulations (DNS). Results show that for high frequencies and flapping amplitudes, the robot is able to produce lift up to 600 mg. For these specific cases, it can be observed from HSV that aerodynamic and inertial forces are sufficient for the wing to bend chordwise (around the hinge) and operate at angles of attack close to 45°. Chordwise bending decreases as the flapping frequency decreases resulting in a frequency threshold below which no lift is being generated. In addition, results show that loads derived from PIV match those obtained from balance measurements and can therefore be used to measure even lower lift production where balance measurements are not accurate. Finally, DNS helps reveal prominent aerodynamic mechanisms at play.

*Email address(es): thierry.jardin@isae.fr

1 INTRODUCTION

The recent advent of micro and nano-technologies have paved the way for the development of extremely small-sized robots capable of flying in confined environments. At these scales, typically on the order of the centimeter, aerodynamic performance of conventional rotary wing concepts drastically decreases due to increased flow viscous effects with respect to inertial effects, i.e. due to extremely-low-Reynolds-number effects. Consequently, because nano-flying robots operate on the same scale of insects, the flapping wing concept has long been proposed as a prospective alternative to the conventional rotary wing concept [1]. However, despite extensive researches over these two or three past decades, there is no compelling evidence that flapping wings are more aerodynamically efficient than rotary wings, even at very low Reynolds numbers. This lack of evidence can partly be explained by our inability to optimize flapping kinematics due to the very large parameter space that needs to be explored. Similarly, while some studies indicate that flapping wings may produce larger lift than rotary wings, there is no compelling evidence that this is intrinsically valid since comparisons are usually performed under restrictive asumptions (e.g. at same operating frequency). Further investigations on flapping wings should thus be conducted to clarify the picture. Yet, apart from pure aerodynamic considerations, [2] suggested that flapping wing concepts could be more efficient than rotary wing concepts at very small scale (fruit fly scale) when taking into account the actuator performance (oscillatory versus rotary). In addition, flapping wings could potentially be conducive to both enhanced maneuverability and aero-acoustic stealth.

In this context, we report on the development of a 200 mg bio-inspired, flapping wing robot. The flapping motion of the wings is driven by a continuous electrostatic motor connected to a simple rod-crank system. The pitching motion (about the wing's leading edge) is passive and results from the dynamic response of the wing structure to aerodynamic and inertial forces. Wings with different structures are manufactured and tested on three different configurations of the rod-crank system with flapping amplitudes 60, 90 and 120°. The aerodynamic lift generated by the flapping wings is assessed for flapping frequencies up to 30 Hz using both balance and particle image velocimetry (PIV) measurements. High-speed camera visualizations are used to extract wing motion and deformation (i.e. flapping, passive pitch and spanwise bending) during the flapping cycle, which is then used for direct numerical simulations (DNS) of the flow past the flapping wings.

Overall, this study shows that the present robot can generate lift about three times its target weight when operated at the largest flapping amplitudes and frequencies and with a wing structure that allows for sufficient pitch amplitude. In addition, it is shown that lift obtained from both balance and PIV measurements match within reasonable accuracy, suggesting that PIV measurements could be used for lift evaluation at even lower scales, i.e. where balances are not accurate. Lift obtained from numerical simulations also reasonably agrees with experimental data, which allows to correlate prominent flow features with aerodynamic performance.

2 Materials and methods

The flapping robot consists of two wings connected to an electrostatic motor via a rod-crank system. The wings are manufactured out of Titane TA6V (chemical cutting procees for structure) and Parylene (chemical vapor deposition method for the membrane). They weight approxiamtely 10 mg with chord and span dimensions of approximately 1 and 3 cm respectively. A 25 μm thick hinge allows the main part of the wing to passively pitch (about its leading edge) with respect to a second part that is rigidly fixed to the rod-crank system. The electrostatic motor is manufactured out of silicon (SilMach's PowerMEMS patented solution). It weights less than 40 mg and is able to operate at frequencies up to a few Hertz. For experimental tests described below, the motor is replaced by a Faulhauber commercial brushless DC-Servomotor (5g weight) to allow operating frequencies up to 30 Hz (figure 1). Finally, the rod-crank system is manufactured out of Titane TA6V and PEEK to minimize both mass and friction, for a weight below 70 mg. For testing purposes, three different rod-crank arrangements are considered, yielding flapping amplitudes of 60, 90 and 120°. The final weight of the protoype is approximately 200 mg.

2.1 Load measurements

The lift produced by the flapping wings is first measured using a Kern EMR 1000-2 balance with a precision of 10 mg and a maximum range of 1000 g. The robot is mounted upside-down on a 3D-printed support to avoid any bias resulting from ground effects (i.e. interaction between the wake of the flapping wings and the ground), see figure 1. The balance provides a steady value for lift. In what follows, reported values for lift are obtained by averaging ten consecutive measurements. Error bars are derived from the maximum deviation of each measurement to the averaged value, corresponding to an uncertainty of $\pm$20 mg.

2.2 PIV measurements

In addition to direct load measurements from balance, stereoscopic particle image velocimetry (PIV) measurements in the wake of the robot are performed (figure 2) to indirectly extract forces from flow velocity integration on a control volume.

The environing fluid is seeded with oil particles. A Nd:YAG laser illuminates particles in a 3 mm thick sheet located 40 mm below the robot. Two high-definition FlowSense EO 16M cameras (4872 px $\times$ 3248 px resolution) synchronized with the laser acquire double-framed images of particles in the laser sheet at a frequency of approximately 2 Hz using Dantec DynamicStudio . The time step between two frames depends on the operating frequency of the robot (which is related to the wake velocity) and is on the order of a few tens of μs. It is chosen to ensure that enough particles remain within the 3 mm thick laser sheet between two frames. The angle between the laser sheet and the cameras is close to 45° to maximize measurement accuracy on both in-plane and out-of-plane displacements.

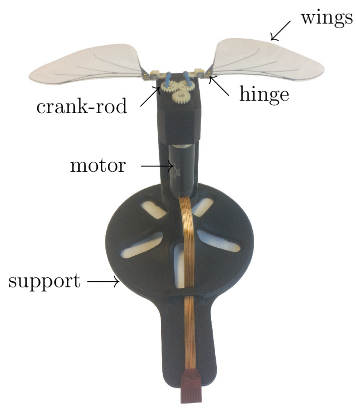


Figure 1: The nano-flying robot mounted (upside down) on the balance support.

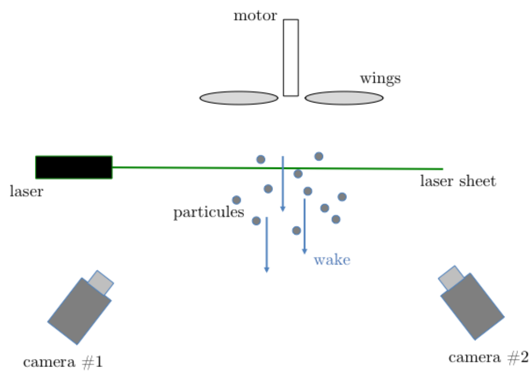


Figure 2: Scheme of the experimental setup for PIV measurements.

After removing background noise, two frames are cross-correlated to obtain an instantaneous, two-component pixel displacement field for each image of each camera. Multi-pass cross-correlation is performed using interrogation windows with decreasing size, from 64$\times$64 to 16$\times$16 pixel square.

Corresponding instantaneous pixel displacement fields from the two cameras are then used to reconstruct a two-dimensional, three-component displacement field. Velocity fields are deduced from displacement fields knowing the time step between two frames.

For each case, 500 instantaneous velocity fields are averaged to obtain a mean (statistically converged) flow field which is then used to extract lift from velocity integration.

2.3 Load extraction from PIV

Instantaneous aerodynamic forces experienced by a body can be extracted by integrating momentum equations on a control volume enclosing the body [3, 4]. This typically requires knowledge of acceleration flow fields inside the control volume and velocity and pressure fields on control surfaces. While time-resolved volumetric PIV measurements can provide such data, it remains highly challenging and restricted to some specific cases. When mean aerodynamic forces are considered, the approach reduces to the integration of velocity and pressure forces on control surfaces [5]. In addition, if the control surfaces are suffciently far away from the body, the sole integration of the velocity field in the wake of the body can provide reasonable estimates of the mean force perpendicular to the measurement plane. Hence, the lift generated by the robot is here derived from integration of the mean vertical velocity field in the PIV measurement plane. It is important to mention that, in this study, while upstream and side surfaces can be taken sufficiently far away the robot, the downstream surface (measurement plane) is rather close to the wings such that the previous hypothesis is not fully verified. That is, integration of pressure and turbulent stresses on the downstream surface would theoretically yield more accurate results. However, pressure and turbulence stresses are known to be strong sources of noise that can potentially affect forces estimation, in a more significant way than simply neglecting their integration.

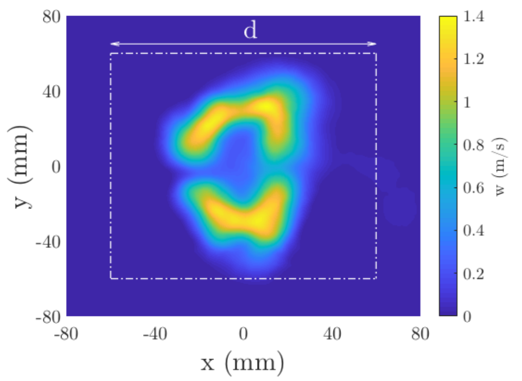


Figure 3: Contours of vertical velocity obtained from PIV measurements for wings operated at a frequency of 25 Hz and with an amplitude of 120°.

Figure 3 shows the magnitude of the vertical velocity component in the measurement plane for wings operated at a frequency of 25 Hz and with an amplitude of 120°. A square control surface centered on the center of the wake is used. Its boundaries are depicted in white. Time-averaged lift is deduced from integration of the vertical velocity field over this control surface. Figure 4 shows the time-averaged lift as a function of the charactestic dimension of the control surface, d. It is shown that the extent of the measurement plane is enough to ensure that lift is converged with respect to the size of the control surface (it staurates for $d > 100$mm), i.e. the side boundaries are sufficiently far away from the wake.

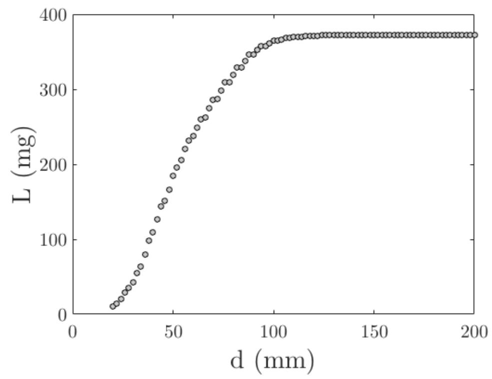


Figure 4: Lift as a function of the characteristic dimension of the control surface.

2.4 High-speed visualizations

A X-Stream Vision 3 IDT high speed camera (1080 px $\times$ 1024 px resolution) placed perpendicularly to the mean wing path (i.e. to the PIV measurement plane) acquires images of the wings at frequencies up to 600 Hz. Figure 5 shows a sequence obtained during one stroke of the flapping motion for wings operated at a frequency of 25 Hz and with an amplitude of 120°. Assuming that wing pitch motion is a solid rotation of the membrane around the hinge, one can estimate the instantaneous angle of attack of the wing, α, by measuring the projected wing chord onto the image, $c \cos \alpha$ (c being the wing chord). Furthermore, one can measure the effective flapping angle of the wing which is here taken as the angular displacement of the wing tip. It can be seen that because the wing bends spanwise, the effective flapping angle is larger than the flapping amplitude of the rod-crank system. It should be noticed that a stroke starts when the angle produced by the rod-crank system is maximum, which does not necessarily corresponds to maximum angle of attack and/or maximum effective flapping angle precisely because of fluid-structure interactions (passive pitch and spanwise bending). Two 150 W Dedolight Halogen light spots are used to continuously illuminate the wings and image acquisition is performed with an exposure time of 300 μs. Filters are then used to enhance the image contrast and extract the wing geometry.

Figure 6 displays the instantaneous angle of attack, α, and effective flapping angle, ϕ, as a function of time (non-

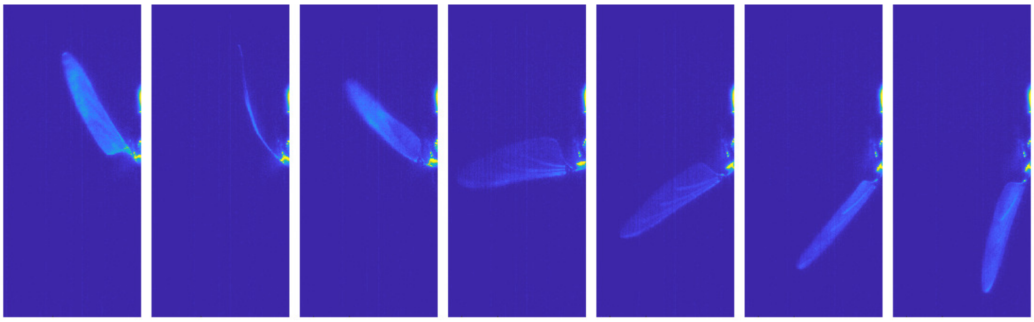


Figure 5: Visualizations of a single wing at 7 instants during one flapping stroke for flapping amplitude $\Delta\phi = 120°$ and frequency $f = 25$Hz.

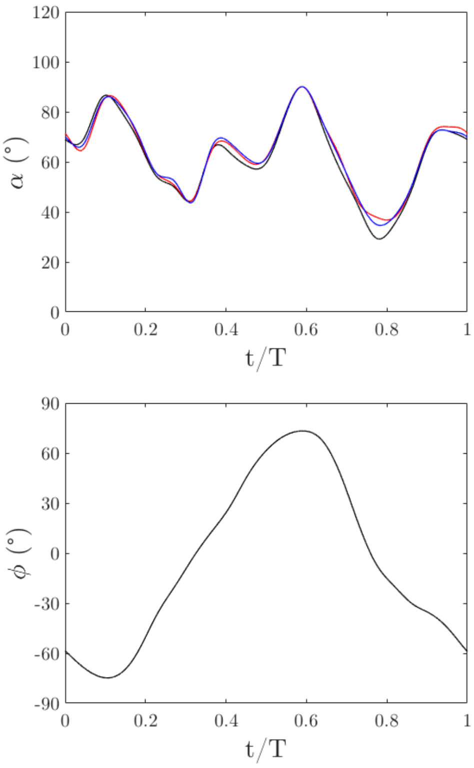


Figure 6: Instantaneous angle of attack and effective flapping angle as a function of non-dimensional time, extracted from HSV for flapping amplitude $\Delta\phi = 120°$ and frequency $f = 25$Hz. α is shown for three consecutive flapping cycles.

dimensionalized by the flapping period T), for three consecutive flapping cycles. Kinematics are shown for the case considered in figure 5. It can be seen that the signal is reasonably repeatable from one cycle to another. In addition, as previously explained, it can already be noticed that the angle of attack does not pass through 90° at $t/T = 0$, i.e. passive pitch induces a phase-lag with stroke reversal. Similarly, the effective flapping angle is not maximum at $t/T = 0$ because of spanwise bending.

2.5 *Direct numerical simulations*

Instantaneous angle of attack and flapping angle are then implemented in a code that directly solves the incompressible form of the Navier-Stokes equations on a finite volume mesh enclosing one wing. An overset grid approach is used that allows the wing mesh to move following prescribed flapping and pitching motions within a stationnary background mesh. The trimmed mesh consists of 5 million hexahedral cells (2.25 million for wing mesh and 2.75 million for background mesh) enclosed within a cylindrical domain of radius 30R and height 40R. The typical cell size at the wing surface and in the close wake is equal to $c/100$ and $c/50$ respectively. The boundary conditions upstream and downstream of the wing are implemented as pressure Dirichlet conditions while the periphery of the cylindrical domain is defined using a slip-wall condition. The wing is modelled as a non-slip surface. Wing mesh is moved with a time step that is 400 times smaller than the flapping period. Both spatial and temporal discretizations are achieved using second-order schemes. Momentum and continuity equations are solved in an uncoupled manner using a predictor-corrector approach.

Preliminary tests showed that aerodynamic loads are converged with respect to both spatial and temporal discretizations. In addition, 10 periods were simulated to allow for initial transients to sufficiently decay.

3 Results

3.1 Mean lifting force

Figure 7 shows the lift (in mg) generated by wings flapping with frequencies up to 30 Hz and flapping amplitudes 60, 90 and 120°.

First, it can be observed that reasonable agreement (within the error bar) is achieved between loads obtained from balance and PIV measurements. In addition to cross-validating both approaches, this indicates that load estimation from PIV measurements can be used for non-intrusive measurements and suggests that the approach could be suited to even weaker magnitudes of lift (e.g. smaller robot dimensions) for which typical balance ranges are not adapted. Lift obtained from DNS overestimates by 12.6% experimental values. This overestimation may partly arise from the solid rotation asumption (the numerical simulation does not account for spanwise wing deformation), inaccuracy in kinematics extraction from HSV and in the fact that only one wing is being simulated.

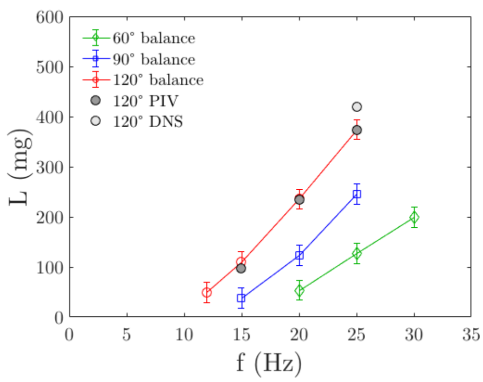


Figure 7: Lift force generated by the flapping wings as a function of the flapping frequency f for three amplitudes $\Delta\phi = 60, 90$ and $120°$. Comparison between results obtained experimentally (balance and PIV) and numerically (DNS).

Second, it can be observed that constant-amplitude lift curves do not approach pure quadratic functions of the frequency. For prescribed angles of attack, a first approximation is to consider a constant lift coefficient C_L such that lift is a quadratic function of both flapping amplitude and frequency $L = \frac{1}{2}\rho A C_L (2\Delta\phi R f)^2$, where ρ is the fluid density, $\Delta\phi$ is the flapping angle swept by the wing and A and R are the wings area and radius respectively. This relationship helps explain the observed increase in lift with both amplitude and frequency but does not allow for the observed lift-to-frequency relationship (alternatively lift-to-amplitude relationship) to be predicted. Here, passive pitch due to fluid-structure interactions imply that the instantaneous angle of attack of the wing highly depends on both flapping amplitudes and frequencies. In particular, operating frequencies lower than 10 Hz did not allow for sufficient pitch motion such that no lift was generated, whatever the flapping amplitude.

Figure 8 shows the lift generated under similar operating conditions but with a thicker, 50 μm thick hinge. Again, reasonable agreement is observed between direct balance measurements and PIV measurements, which supports previous conclusions. Furthermore, it is observed that no significant differences exist with results obtained with a 25 μm thick hinge. In other words, effects of hinge flexibility on fluid-structure interaction processes appears to be of second order as compared to aerodynamic forces and wing weight and inertia.

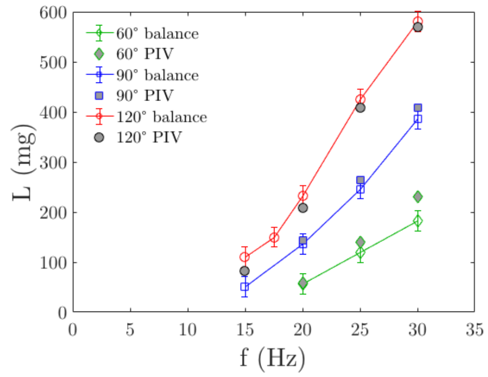


Figure 8: Lift force generated by the flapping wings as a function of the flapping frequency f for three amplitudes $\Delta\phi = 60, 90$ and $120°$. Results obtained experimentally (balance and PIV) with a 50 μm thick hinge.

Overall, it is demonstrated that the robot can generate enough lift force (i.e. superior to its weight) for sufficiently large flapping amplitude and frequency, typically $\Delta\phi > 90°$ and $f > 20$ Hz. Lift up to nearly three times the target weight is obtained for $\Delta\phi = 120°$ and $f = 30$ Hz.

3.2 Passive pitch

The effect of frequency and amplitude on passive pitch motion is highlighted on figure 9 which shows the instantaneous angle of attack α (phase-averaged over 2 consecutive strokes) extracted from high-speed visualizations. α is here measured with respect to the wing path direction. Three cases with $\Delta\phi = 120°$, $f = 25$ Hz; $\Delta\phi = 120°$, $f = 15$ Hz and $\Delta\phi = 60°$, $f = 25$ Hz are compared. It is observed that the wing operates at relatively similar angles of attack for $\Delta\phi = 120°$, $f = 15$ Hz and $\Delta\phi = 60°$, $f = 25$ Hz explaining comparable values of averaged lift for these two cases (see figure 8). Conversely, the wing operates at lower angle of attack in the $\Delta\phi = 120°$, $f = 25$ Hz case, which explains a larger production of lift for this case. Mean values of α are 66, 76 and 77° for cases $\Delta\phi = 120°$, $f = 25$ Hz; $\Delta\phi = 120°$, $f = 15$ Hz and $\Delta\phi = 60°$, $f = 25$ Hz, respectively.

In addition, it is shown that all α curves exhibit a complex trend with multiple harmonics. These complex trends, together with a limited passive pitch motion (which does not allow for sufficiently low, sustained values of α to be obtained over one stroke), are not optimal in terms of aerodynamic performance. Thus, although sufficient lifting force is provided, further improvements can be achieved through modifications of the wing structure to promote passive pitch motion.

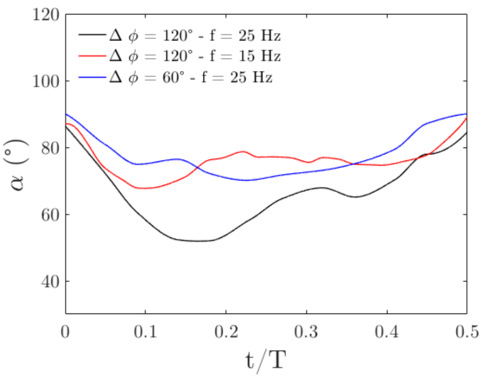


Figure 9: Comparison of instantaneous angles of attack extracted from HSV for cases with $\Delta\phi = 120°$, frequency $f = 25$Hz; $\Delta\phi = 120°$, frequency $f = 15$Hz and $\Delta\phi = 60°$, frequency $f = 25$Hz.

3.3 *Correlation between instantaneous lift and flow physics*

Wing kinematics extracted from high-speed visualizations (figure 6) are implemented in a flow solver to predict the instantaneous force generated by a single wing (with 25μm thick hinge) flapping at a frequency of 25 Hz and with an amplitude of 120°.

Figure 10 displays the instaneous lift generated by the wing over a flapping period. The curve exhibits low levels of lift around $t/T = 0$, 0.5 and 1, i.e. at stroke reversal where the wing velocity approaches zero. Immediatly after stroke reversal, two lift peaks ($t/T \approx 0.15$ and $t/T \approx 0.65$) are generated despite the relatively high angle of attack (figure 6). These peaks most presumably result from strong added mass effects associated with the severe acceleration of the wing. It was observed from figure 5 that the wing bends along its span as it changes direction, and then strongly accelerates as a response to internal stresses. Note that while wing bending is not directly taken into account in the numerical simulation, the resulting acceleration is intrinsically contained in the wing kinematics implemented in the flow solver. After these peaks, the angle of attack decreases to values that favor lift production (approximately 45 and 30° in the first and second halft stroke respectively) through the generation of a leading edge vortex.

Figure 11 shows iso-surfaces of Q-criterion during one stroke. Overall, the flow is characterized by multiple small

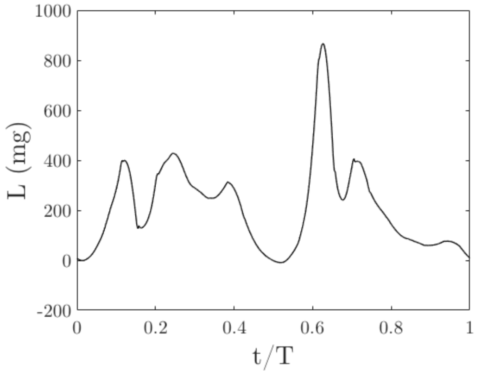


Figure 10: Instantaneous lift obtained numerically for wings operated with flapping amplitude $\Delta\phi = 120°$ and frequency $f = 25$Hz.

scale structures on the outboard portion and in the wake of the wing. These structures strongly interact with the wing at stroke reversal (first and second snapshots). As the wing further revolves about its root, a leading edge vortex (LEV) is formed, which exhibits a dual shape structure [6] (third snapshot). The LEV, a principal contributor to lift generation on flapping and revolving wings [7], develops from the root to a radial station where it bursts into small scale structures. In the inboard region, the LEV appears to remain attached to the wing for the whole stroke. It eventually lifts off the wing surface at the next stroke reversal (last snapshot).

As such, DNS shows that despite the chaotic trend and the relatively high value of angle of attack during one stroke, a relatively stable LEV develops on the wing, which favors lift production. In addition, spanwise bending induces strong acceleration at stroke reversal, which also contributes to lift generation through large-amplitude lift peaks.

4 Conclusion

We reported on the development of a 200 mg, bio-inspired nano flying robot. We addressed the main issues related to the development of extremely small-sized and light-weight vehicles in terms of manufacturing materials and processes as well as performance evaluation. Using direct balance measurement and indirect lift evaluation by particle image velocimetry, we showed that the robot was able to generate lift about three times its weight when the wings were operated at the largest frequency and amplitude tested. In addition, we provided insight into the instantaneous generation of lift by measuring the wing kinematics (with passive pitch motion) and implemented them in a flow solver, which further gives access to the full three-dimensional flow field around the wings. It was suggested that although the wings were able to generate a sustained LEV throughout the stroke and could benefit from spanwise bending at stroke reversal to produce lift, further imporvements could be achieved in terms of aerodynamic performance through modification of the wing structure and enhanced passive pitch motion.

Acknowledgements

The authors would like to thank DGA (Délégation Générale de l'Armement) for funding this work under the 'Libellule' III project.

References

[1] D. Floreano and R. Wood. Science, technology and the future of small autonomous drones. *Nature*, 521(7553):460, 2015.

[2] E.W. Hawkes and D. Lentink. Fruit fly scale robots can hover longer with flapping wings than with spinning wings. *Journal of the Royal Society Interface*, 13(123):20160730, 2016.

[3] D.F. Kurtulus, F. Scarano, and L. David. Unsteady aerodynamic forces estimation on a square cylinder by tr-piv. *Experiments in Fluids*, 42(2):185–196, 2007.

[4] L. David, T. Jardin, and A. Farcy. On the non-intrusive evaluation of fluid forces with the momentum equation approach. *Measurement Science and Technology*, 20(9):095401, 2009.

[5] B.W. van Oudheusden, F. Scarano, E.W. Roosenboom, E.W. Casimiri, and L.J. Souverein. Evaluation of integral forces and pressure fields from planar velocimetry data for incompressible and compressible flows. *Experiments in Fluids*, 43(2-3):153–162, 2007.

[6] Y. Lu, G.X. Shen, and G.J. Lai. Dual leading-edge vortices on flapping wings. *Journal of Experimental Biology*, 209(24):5005–5016, 2006.

[7] M.H. Dickinson, F.O. Lehmann, and S.P. Sane. Wing rotation and the aerodynamic basis of insect flight. *Science*, 284(5422):1954–1960, 1999.

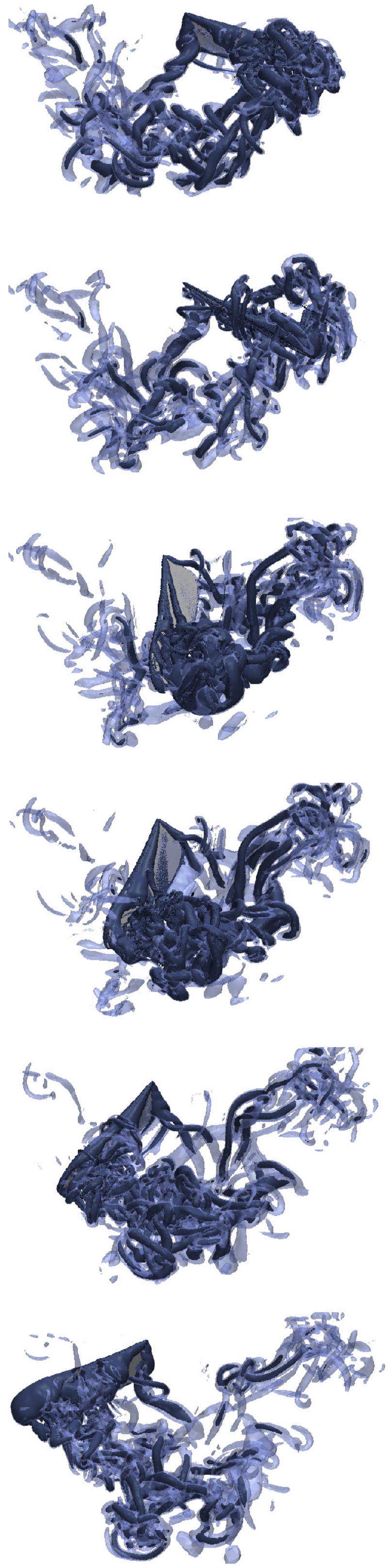


Figure 11: Sequence of Q-criterion isosurfaces obtained from DNS for flapping amplitude $\Delta\phi = 120^\circ$ and frequency $f = 25$Hz.

Controlling The Aerial Posture of a Flapping-wing Micro Air Vehicle by Shifting Its Centre of Gravity

Gakuki ISE[1],Daiki YODA[1], Chinami TAKIGUCHI[1], Naoya SHIGEMATSU[1], Tomohiro SHIRATORI[1], Hisami NAKANO[1], Woojae JEON[1], Taichi NOZAWA[1], Kei IWASHIRO[1], Takumi NISHINO[1], Toshitatsu MUNAKATA[2] and Takanobu WATANABE[1]
1Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan
2Tyoukogata Hikoutai lab., http://mayoneko.wixsite.com/microslowfly

ABSTRACT

We have developed a Flapping Wing Micro Air Vehicle (FW-MAV) equipped with a mechanism to fly in 2 modes: hovering and horizontal flight. The mechanism consists of a servo motor with battery mounting stay. The battery moves between positions of lying above and below the wings, thereby the centre of gravity (COG) shifts around over the wings. By this feature, the FW-MAV can change the flight posture from aerial-stop posture to the horizontal-flight posture. This flight mode transition was recorded by a high-speed camera and motion capture system.

1. INTRODUCTION

For the purposes of video shooting, search-and-rescue operation, communication, agriculture and environment preservation, various types of Micro Air Vehicle (MAV) have been developed [1-3]. Quadrotor type MAVs enable hovering flight, which is suitable for close surveillance operation. Fixed-wing type MAVs are suitable for long distance flight.

Some missions require the capability to take the two different flying modes, and several ideas have been proposed for this purpose. Green et al. developed a fixed wing airplane which can fly vertically by moving ailerons and elevators [4]. They solved the lack of stability in vertical flight using autonomous software control which is said to be better than a matured pilot. However, it could only keep flying vertically up to 90 seconds.

Flapping-Wing Micro Air Vehicle (FW-MAV) is an alternative solution, since it has the potential to realize both vertical and horizontal flights. Furthermore, the weight scalability, the camouflage ability and the high-mobility flight capacity of FW-MAVs are advantageous and attracting a lot of interests [5]. In our previous study, we proposed a concept to realize both vertical and horizontal flights by changing aerial posture MAV's with the mechanics of centre-of-gravity (COG) shift. [6] Koopmans et al. investigated the applied forces on the flying FW-MAV and the relationship between COG and the flight manner [7]. They further implemented an FW-MAV with a shift mechanism of gearbox position to change the COG, and thus the FW-MAV takes the ability of vertical and horizontal flight modes.

In this paper, we demonstrate a new FW-MAV named "Wifly" equipped with a simple mechanism to change the areal posture. This paper is organized as follows. Section 2 elaborates on the overall design of "WiFly." In Section 3, we will explain the mechanism of COG shift to change the aerial posture. Sections 4 and 5 shows the motion capture analysis to quantitatively evaluate the effect of the COG shift mechanism.

2. OVERVIW OF "WiFly"

Figure 1 shows an overall picture of WiFly, which comprises of a gear box, flapping wings, tail wings,

*contact email: as-gaku-gaku@akane.waseda.jp

tail rotor, micro-computer chip for controlling actuators and Centre of Gravity Shift (COGS) mechanism. Figure 2 shows a dimensional outline drawing together with components layout. The components are settled on a frame made of carbon-fiber shafts. Four carbon-fiber shafts are arranged in parallel to form a rectangular cross-section frame. The COGS mechanism is settled on the upper side of the wings, across the centre of which the lithium polymer battery pass through. To optimize the weight balance in the hovering mode, the gearbox is installed as it is shown in Figure 1 so that the wings are set on the bottom side of the frame. We employed "Lazurite Fly" as the control circuit board. This is a prototype product offered by LAPIS semiconductor Co., Ltd. It is equipped with several sensors and communicates with a controller via a 920 MHz wireless connection. The tail rotor adjusts the yaw moment by pushing the tail right or left. The body weight including the battery weighs only about 32 g.

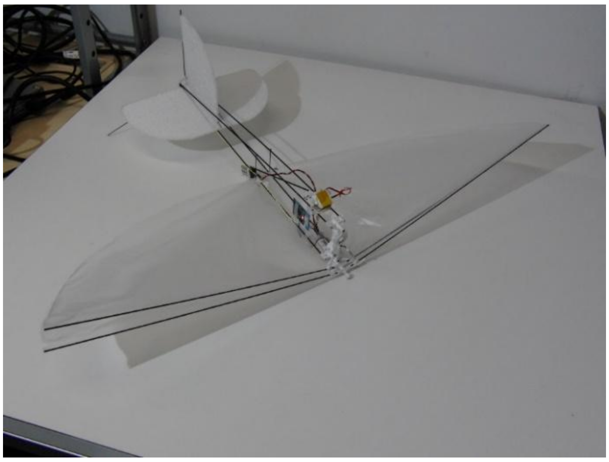


Figure 1 - WiFly

Weight	32g
Wing span	62 cm
Frame length	39 cm
Battery	3.7 V LiPo battery

Table 2 – Specifications of WiFly

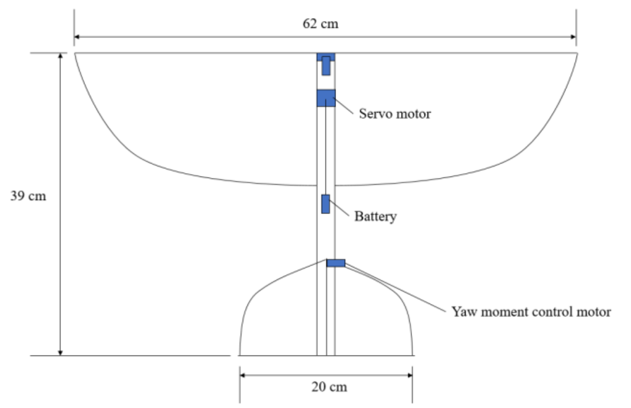


Figure 2 – Dimensional outline of WiFly and components arrangement

3. COGS MECHANISM

The appropriate positioning of COG is the key to stabilize the aerial posture of FW-MAV like those of normal aircrafts. This is true not only in the horizontal flight, but also in the hovering flight. The COGS is realized by the change in the mass distribution in the FW-MAV. Koopmans et al. developed a FW-MAV with a function to change the aerial posture by sliding the wing and gearbox, which are the heaviest parts in the FW-MAV [7]. The slide mechanism was realized by modifying a micro linear servo motor.

In our developed COGS mechanism, the battery is swinged between above and under the wings, and it realizes the seamless change between hovering flight and horizontal flights modes.

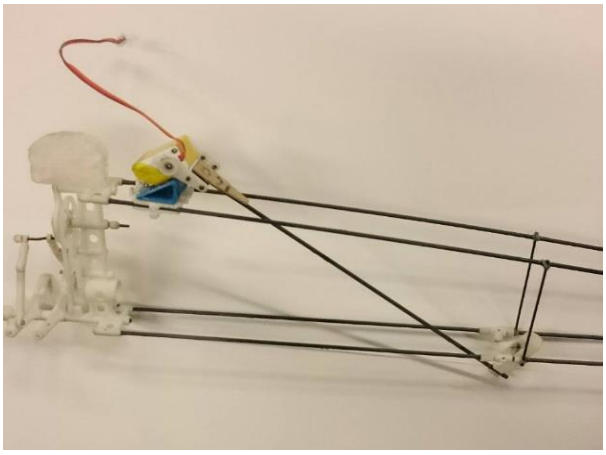


Figure 3 - The CPGS mechanism

Figure 3 shows the COGS mechanism. The servo motor rotates 75 degrees, and the battery moves vertically across through the carbon-shaft frame. Figure 4 shows the definition of internal axes of WiFly. The components are arranged symmetrically the z-x plane, so that the COG moves in this plane. We measured the trajectory of COG transition by hanging the FW-MAV from different points. Figure 5 shows the photos of the hanging measurement, and the obtained COG trajectory is shown in Figure 6. As indicated in Figure 6, the COG shifts to above the wings and backward, as the servo motor rotates from 0 to 75 degrees.

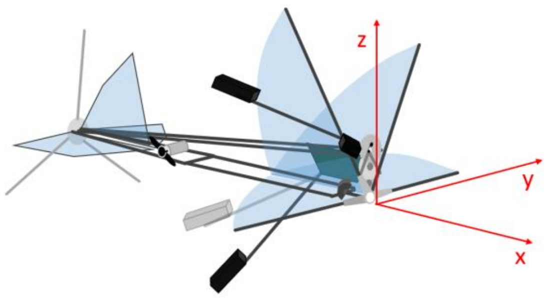


Figure 4 - Definition of internal axes of WiFly

Figure 5 - images of hanging experiments to measure COG

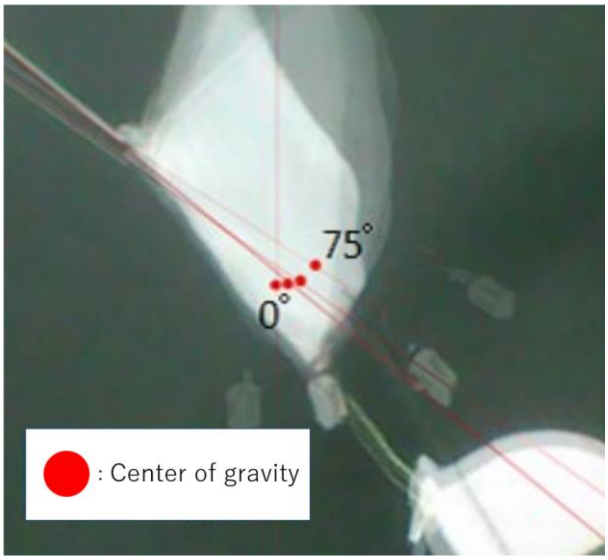


Figure 6 - The range of COG motion

4. FLYING EXPERIMENTAL

To demonstrate the effect of the COGS mechanism, we performed motion-capture experiments. Figure 7 shows an image of the experiments. The flight area is 6 x 5 meters in depth and width, and about 3.5 meters in height. We recorded the trajectory from the taking off from a vertical starting position into the horizontal flight. The WiFly was set about 30 cm above the floor in order that the motion-capture system can track the taking off. Detection/tracking markers were stuck on the gearbox and tail wing.

Figure 7 - Actual image of motion-capture Experiment

In the second experiment, we recorded the horizontal flight. The detection markers were placed on the same positions as the first experiment. WiFly was launched from hand with about 45 degrees pitch angle, stand-bying with the flapping motion on the hand. The flapping power is fixed at about 80 % of the maximum, which is the best condition to keep the stable horizontal flight. This percentage was the best-practice to keep the stable horizontal flight.

5. RESULTS AND DISCUSSION

Figure 8 shows a series of photos taken by high-speed video recording in which WiFly changed its flight posture. In the first and second shots, the battery is shaded by the wings because it is located above the wings. In the third and fourth shots, the battery swung down and can be seen in photos and the posture is changed. Figure 9 shows the recorded trajectory and areal posture by the motion capture system in the sequence of the take-off, hovering flight and horizontal flight. The coordinate is shown in meter scale. WiFly successfully took off, moved forward slightly and then continued ascending while shaking its tail. After it arrived at the peak altitude, the flight mode is changed into horizontal. WiFly once lost altitude just after the shift of the flight posture, because flapping power is temporarily suppressed. This operation is required to switch the flight modes. Figure 10 shows a trajectory of a hovering flight of about 14 seconds. WiFly kept the constant altitude slightly with a periodic swaying motion. The swaying motion induces the undesired horizontal migration. The swaying motion can be cancelled by yaw control rotor, which is left for future work.

Figure 11 shows a trajectory of horizontal flight . Some data points are dropped, where the motion capture failed to track the position of a few markers. Just after the launch, WiFly was accelerated ahead as it bows downwards slightly. As it gained speed, the trajectory is gradually elevated.

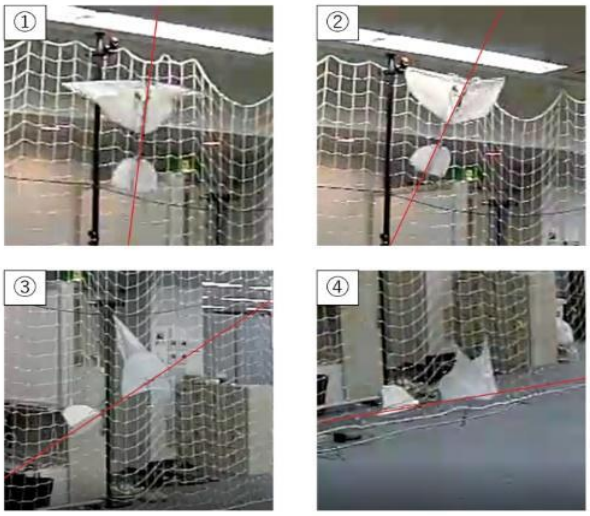


Figure 8 - Sequence of images of transition from vertical posture to horizontal posture

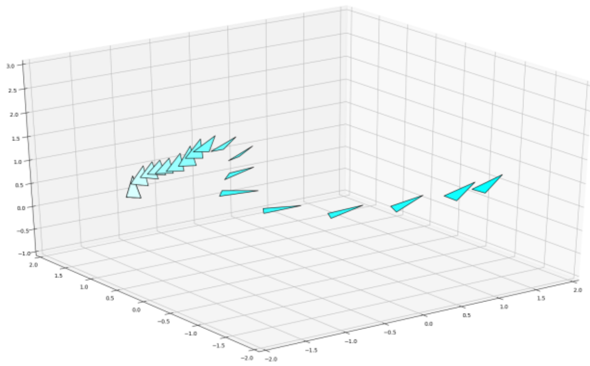


Figure 9 - Trajectory of a navigation from the taking-off to horizontal flight

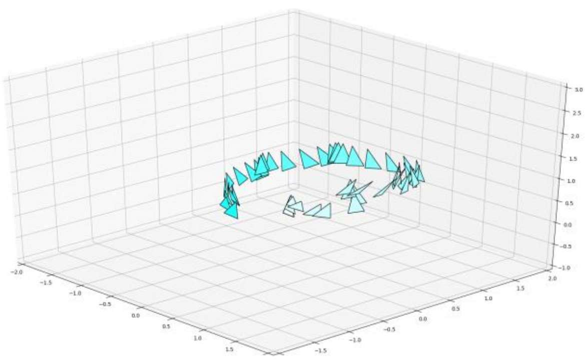


Figure 10 - Trajectory of a hovering flight

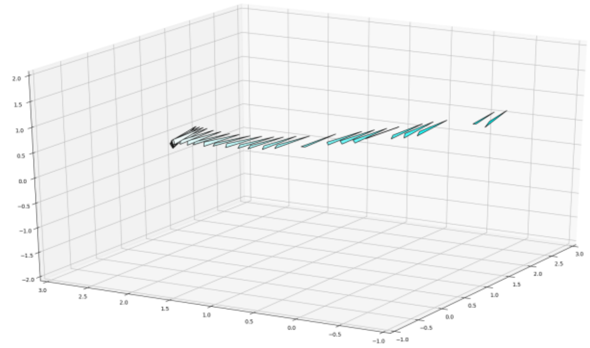


Figure 11 - Trajectory of a fast forward flight

5. FUTURE WORK

The periodical swaying motion during hovering flight shown in Figure 10 is undesired, because it sometimes causes the FW-MAV to lose the balance. This motion could be cancelled by the yaw-moment control rotor. Furthermore, it is urgently required to implement a flight assistance program to change the flight modes, because flapping power and the angle of servo should be controlled precisely at the same time.

6. SUMMARY

We have developed a flapping wing MAV with a mechanism to shift the centre-of-gravity, which comprises a servo motor and battery mounting stay. It moves the battery between lying above and below the wings. It was demonstrated by motion capture experiments that the MAV took-off vertically and hovered at a constant altitude, and successfully changed the aerial posture into horizontal flight.

ACKNOWLEDGEMENTS

This research was supported by Aerosense Inc. and the START program provided by The Japan Science and Technology Agency. This study was conducted as a part of the project (14P53) of the Research Institute for Science and Engineering. We thank Mr. N. Saito of LAPIS semiconductor Co., Ltd. for providing us "Lazurite Fly."

REFERENCES

[1] FLOREANO, Dario; WOOD, Robert J. Science, technology and the future of small autonomous drones. Nature, 2015, 521.7553: 460.

[2] TOMIC, Teodor, et al. Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. IEEE robotics & automation magazine, 2012, 19.3: 46-56.

[3] QUIGLEY, Morgan, et al. Target acquisition, localization, and surveillance using a fixed-wing mini-UAV and gimbaled camera. In: *Robotics and Automation, 2005.*

[4] GREEN, William E.; OH, Paul Y. Autonomous hovering of a fixed-wing micro air vehicle. In: Robotics and Automation, 2006.

[5] HAN, Jae-Hung; LEE, Jun-Seong; KIM, Dae-Kwan. Bio-inspired flapping UAV design: a university perspective. In: *Health Monitoring of Structural and Biological Systems 2009*.

[6] Daiki, Y., et al., "Development of Flapping Aerial Robot with Movable Gravity Center", The Robotics and Mechatronics Conference 2016 in Yokohama, Yokohama, Kanagawa, Japan, 2016

[7] KOOPMANS, J. A., et al. Passively stable flapping flight from hover to fast forward through shift in wing position. *International Journal of Micro Air Vehicles*, 2015, 7.4: 407-418.

Flapping Wing Micro Air Vehicle (FW-MAV) State Estimation and Control with Heading and Altitude Hold

S. Aurecianus[1], H.V. Phan[2], S. L. Nam[1], T. Kang[1]*, and H.C. Park[2]
[1]Department of Aerospace Information Engineering, Konkuk University, Seoul, South Korea
[2]Department of Smart Vehicle Engineering, Konkuk University, Seoul, South Korea

ABSTRACT

In this work, small control board is specifically designed to control the flapping-wing micro air vehicle (FW-MAV) called KUBeetle. In addition, remote control is also custom-built to transmit control command and receive flight data in real time. Then, a method to estimate the attitude and altitude is developed and used to control the behaviour of KUBeetle. The attitude estimation is obtained by combining gyroscope, accelerometer, and magnetometer measurement. While the altitude estimation is obtained by using ranging sensor combined with z-axis accelerometer. FW-MAV is a system with very high vibration noise which are generated by flapping motion and its body. Without using any mechanical damping, the sensors signals are filtered using low pass and Kalman filter. The altitude estimation is evaluated using ball screw driven linear motion guide system. Proportional-Derivative (PD) and proportional-integral-derivative (PID) control with feedback are implemented for attitude and altitude control, respectively. The KUBeetle can demonstrate hover flight and control its attitude and altitude.

1 INTRODUCTION

Since the past few years, flapping-wing micro air vehicle (FW-MAV) has been studied and developed from several aspects [1, 2], such as aerodynamic model, fluid structure interaction, flapping-wing motion mechanism, small and light control board design and navigation and control. In this paper, mainly two aspects of FW-MAV called KUBeetle [3, 4, 5], control board design and control system, are studied. KUBeetle design is inspired from hummingbirds [6] and some insects [7] which have nearly horizontal flapping stroke plane. Its attitude is controlled by changing flapping properties, such as trailing edges at the wing roots and stroke plane, instead of relying on control surface.

Small control board is designed to estimate onboard state and to control KUBeetle. It is equipped with inertial measurement unit (IMU) containing gyroscopes, accelerometers, and magnetometers, and a ranging sensor. Other FW-MAVs, such as nano hummingbird [8], Robobee [9], Delfly [10], and robotic hummingbird [11], also utilize some of these sensors on their control board. Some researches require the use of external equipment [3, 12], such as high-speed camera, to obtain flight information. Or in some other researches the flight data need to be saved on the on-board memory first. Taking this into account, a remote control is specifically designed which not only transmit control commands to the control board, but also receive flight data from the control board during flight. Thus, the response of the FW-MAV can be observed in real time.

Controlling FW-MAV is a challenging task as the system produces high vibration noise, mainly due to the flapping motion, which could disrupt the sensor measurement. The noise could be reduced by attaching mechanical damping on the control

*contact email: tskang@konkuk.ac.kr

board or through filter in the control program [3, 13]. In order to obtain estimation of pitch, roll, yaw angle, and altitude, low pass filters (LPF) and Kalman filters (KF) [14, 15] are used to filter out noises and combine the sensor signals. The altitude estimation is examined using a ball screw driven linear motion guide system which could provide accurate and precise displacement.

On the previous work [3], KUBeetle can demonstrate stable take off and hover flight through proportional-derivative (PD) control with angular rate feedback from gyroscope. In this work, PD control with angle feedback is implemented for attitude control and proportional-integral-derivative (PID) control with altitude feedback is implemented for altitude control. The addition of magnetometer and ranging sensor helps improving KUBeetle heading and altitude control.

The main contributions of this paper are: (1) design of control board and remote control which allows on board control and access to flight data in real time, (2) developing method to filter and estimate attitude and altitude sufficiently without any mechanical damping, specifically for KUBeetle, and (3) developing control method which improves the performance of KUBeetle, especially for heading and altitude control. The rest of the paper is arranged as follows: in Section 2 the brief introduction to the KUBeetle. Then followed by the control board and remote control design in Section 3. The filters and control method used are described in Section 4 and 5, respectively. The experiment setup and result are shown in Section 6 and the last section presents concluding remarks.

2 KUBEETLE

The KUBeetle model has been updated multiple times during development process. The model that is used in this study implements stroke plane change mechanism, while the previous model [3] implements trailing edge change at the wing roots mechanism. Figure 1 shows the KUBeetle model and its main parts. The attitude control is performed through three servos and the altitude control is performed through coreless DC motor. More detailed information regarding KUBeetle model used in this study can be found in [16].

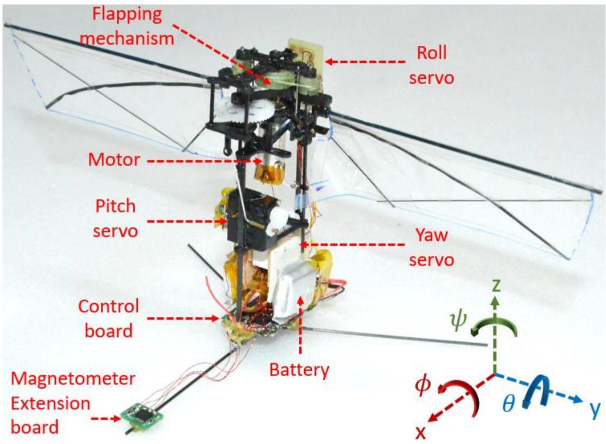


Figure 1 - KUBeetle

3 CONTROL BOARD AND REMOTE DESIGN

The control board is designed so that micro-processor, sensors, and transceiver can be placed in one board to reduce the total weight of the FW-MAV. It consists of a micro-processor STM32L432KC, an inertial measurement unit (IMU) MPU-9250, a ranging sensor VL53L0X, a transceiver nRF24L01+, a motor driver, power regulators, and supporting components, such as resistors and capacitors. The IMU consists of three-axes gyroscope, accelerometer, and magnetometer (AK8963C). Figure 2 shows top and bottom view of the control board. It is made as small and light as possible with some design limitation and restriction. The weight without components is 0.3 g and with components is around 0.9 g. It has maximum width 17 mm and length 23 mm. The board consist of 4 layers with total thickness around 0.4 mm. The IMU is placed at the centre of the board and perpendicular to centre of gravity (CG) of KUBeetle. The ranging sensor is placed at the bottom side, so it shall face the ground after being attached to KUBeetle.

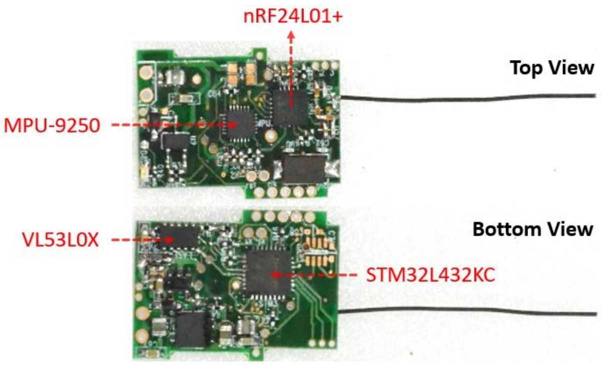


Figure 2 - Control board

The selected micro-processor enables main control loop to have 400 Hz frequency. This frequency still can be increased further with high frequency external oscillator. Currently, the magnetometer on the control board is not used due to magnetic field interference produced by the surrounding high current lines, servos, and motor. Alternatively, an extension board is made specifically for magnetometer so that it can be placed apart from the interference sources. This serve as temporary solution to ensure stable measurement until better solution is found. The ranging sensor uses laser to obtain accurate range measurement. The sensors sampling rate and specification are shown in Table 1. The magnetometer and ranging sensor have slower sampling rates than other sensors but are sufficient enough to control the KUBeetle.

Sensor	Sampling Rate	Output Range	Sensitivity
Gyroscope	400 Hz	± 1000 °/s	32.8 LSB/(°/s)
Accelerometer	400 Hz	± 8g	4096 LSB/g
Magnetometer	100 Hz	± 4800 µT	0.6 µT/LSB
Ranging sensor	25 Hz	0 - 2 m	1 mm/LSB

Table 1 – Sensors sampling rate and specification

Remote control is custom-built with the same transceiver as the control board. It has similar parts as conventional remote control, such as joysticks, potentiometers, and buttons. In addition, IMU is added to allow control by motion when selected. Figure 3 shows the designed remote control. The remote control works as control command transmitter and flight data receiver at the same time. The flight data is sent to computer in real time for observation and recorded. The data rate depends on RF signal stability and data size which can be modified accordingly. With 14 Byte payload data transmission and 115200 baud rates UART/USART communication to the computer, the data rate is around 40 to 50 Hz.

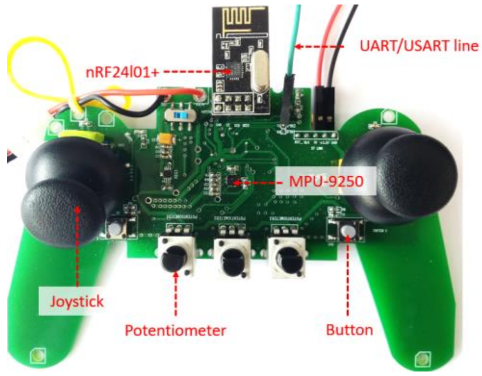


Figure 3 – Remote control

4 FILTERS AND STATE ESTIMATION

The control board is attached firmly at the bottom part of the KUBeetle without any damping material. Consequently, all the noise should be filtered by software in the control board.

4.1 Low pass filter

Infinite impulse response (IIR) LPF is used multiple times throughout the calculation process until the estimation of angle and altitude is obtained. It is used to filter out high frequency noises. The IIR LPF equation can be written as:

$$\hat{\delta}[k] = \alpha\delta[k] + (1-\alpha)\hat{\delta}[k-1] \qquad (1)$$

where $\hat{\delta}$ is filtered signal, δ is unfiltered signal, α is coefficient between 0 to 1, and k is time index. The coefficient value will determine the cut off frequency of the filter and affect time and phase delays of the signal. Firstly, LPF is used to filter the angular rate, acceleration, and magnetic field vector measurement from sensors. Secondly, it is used to filter accelerometer tilt angle signal for pitch and roll which is obtained by inputting filtered acceleration signal to tilt angle equation computed according to the rotation sequence x-yz (roll-pitch-yaw) [17, 18]. Even though the use of low pass filter creates delay, its effect is mitigated by increasing the control loop frequency.

4.2 Kalman filter

Standard discrete Kalman filter (KF) [13, 14] is used to obtain pitch, roll, and altitude estimation. For angle estimation, the state matrix, x_k, state transition model, A, control input model, B, and observation model, H, are defined as:

$$x_k = \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_k \tag{2}$$

$$A = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \tag{3}$$

$$B = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \tag{4}$$

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{5}$$

where θ is estimated angle, $\dot{\theta}_b$ is estimated bias, and Δt is loop period or time interval. The control input of the KF is the filtered gyroscope signal and the measurement of the KF is the accelerometer tilt angle signal. Similar iterative process of KF is used for both pitch and roll. While the estimated yaw angle is obtained using equation derived from inverted rotation matrix of magnetometer with rotation sequence x-y-z (roll-pitch-yaw) and tilt compensation [19].

For altitude estimation, the model is derived from z-axis velocity and distance equation by integrating z-axis acceleration. The state matrix, x_{2k}, state transition model, A_2, control input model, B_2, and observation model, H_2, are defined as:

$$x_{2k} = \begin{bmatrix} V_z \\ h_z \end{bmatrix}_k \tag{6}$$

$$A_2 = \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} \tag{7}$$

$$B_2 = \begin{bmatrix} \Delta t \\ \frac{\Delta t^2}{2} \end{bmatrix} \tag{8}$$

$$H_2 = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{9}$$

where V_z is estimated velocity and h_z is estimated distance on z-axis or altitude. The control input of the system is the gravity compensated filtered z-axis acceleration signal and the measurement of the system is the ranging sensor altitude measurement.

5 CONTROL METHOD

KUBeetle is an inherently unstable system. As preliminary study of KUBeetle attitude control, PD control with angle feedback is implemented to control KUBeetle attitude. The rotation angle is maintained as the initial condition. The control maintains the pitch and roll angle close to zero degree rotation. Furthermore, it maintains the yaw or heading angle close to initial heading, holding the initial heading direction. To hold the altitude of KUBeetle during flight, PID control with altitude feedback is implemented. As it is much easier for KUBeetle to go down than go up, the control output is made much smaller when the altitude error is negative. It can be done by limiting the range of control output or gain scheduling. The altitude control only works when altitude hold command is received. All the control gains are tuned by experiment. Figure 4 and 5 show the control loop diagram for attitude and altitude control, respectively. The set point and offset can be adjusted through remote control.

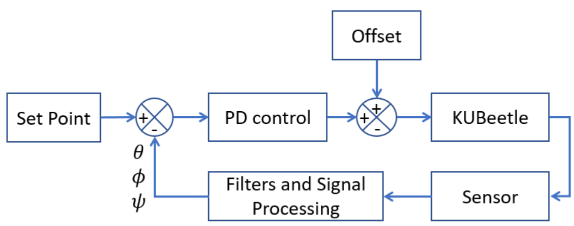


Figure 4 – Attitude control loop

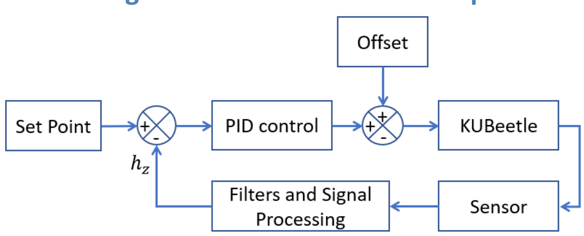


Figure 5 – Altitude control loop

6 EXPERIMENT AND RESULT

6.1 Altitude estimation

The altitude estimation evaluation is conducted using the ball screw driven linear motion system shown in Figure 6. It can change KUBeetle position, higher and lower, at known distance. For convenience, the experiment is conducted using sensors modules and MBED LPC1768 which attached to the moving platform. The displacement distance and ultrasonic sensor measurement are used as comparison. The data is sent to the computer and recorded to be used for data processing and simulation.

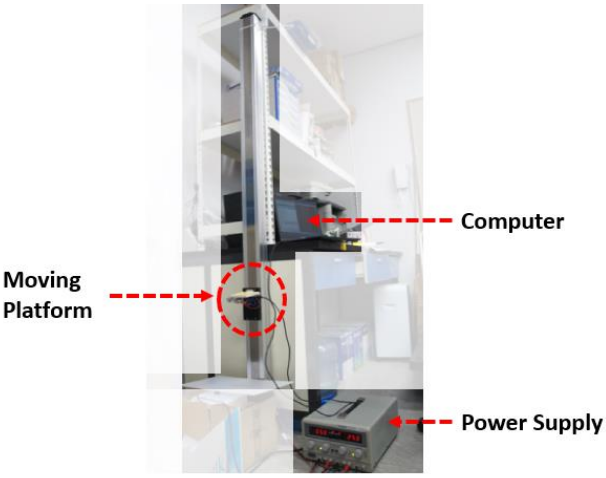


Figure 6 – Linear motion guide system

Figure 7 shows the experiment result for altitude estimation using laser ranging sensor compared with ultrasonic sensor. In this experiment the position of the system is increased by 10 cm every 3s. The laser ranging sensor is quite stable and close to the real height compared to ultrasonic sensor which has a lot more noise as the height increased which caused by echo. While this experiment is conducted without any vibration from flapping motion, the mechanical vibration from the linear motion guide system is much larger than flapping vibration. It can be concluded that the laser ranging sensor is not much affected by the mechanical vibration.

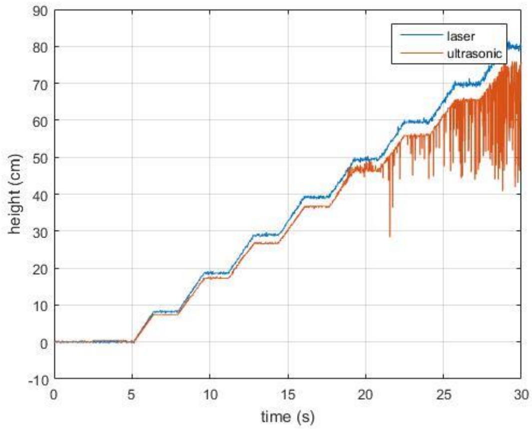


Figure 7 – Altitude estimation

6.2 Flight test

The flight test is conducted in the closed room with no wind. Figure 8 shows the altitude and altitude estimation during hover flight. The attitude estimation values are close to zero with around 10 degree variation which shows that the control is working well and able to maintain the attitude. The red line indicates when the altitude hold is activated. There are some larger altitude oscillation or instability after some time which require more tuning to PID gains to improve the altitude hold performance. This is also affected by the response time of the motor. The altitude graph also shows that the flapping motion and mechanical vibration do not much affect the altitude estimation.

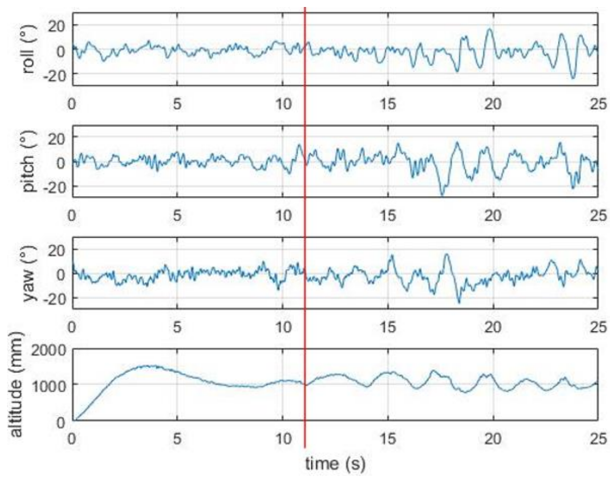


Figure 8 – Attitude and altitude estimation during flight

The snapshot image during flight is shown in figure 9. KUBeetle able to maintain its attitude and altitude with some position changes in x and y axis direction.

Figure 9 – Snapshot image of the KUBeetle during flight

7 CONCLUSIONS

Control board and remote control have been made to control KUBeetle and obtain real time flight data. The data rate is around 40 Hz for 14 Byte transmission payload with 115200 baud rates UART/USART communication. The attitude and altitude of KUBeetle are estimated sufficiently with low pass and Kalman filter without any external mechanical damping. KUBeetle able to demonstrate attitude control, especially heading, with good stability using PD control. The attitude can be maintained within 10 degree in hover flight. While the altitude estimation is quite stable and accurate, the altitude control needs to be improved to have stable altitude.

REFERENCES

[1] J.H. Han, J.S. Lee, and D.K. Kim. Bio-inspired flapping UAV design: a university perspective. Proceedings Volume 7295, Health Monitoring of Structural and Biological Systems, 2009.

[2] M.F.B. Abas, A.S.B.M. Rafie, H.B. Yusoff, and K.A.B. Ahmad. Flapping wing micro-aerial-vehicle: Kinematics, membranes, and flapping mechanisms of ornithopter and insect flight. Chinese Journal of Aeronautics, 29(5):1159-1177, 2016.

[3] H.V. Phan, T.S. Kang, and H.C. Park. Design and stable flight of a 21g insect-like tailless flapping wing micro air vehicle with angular rates feedback control. Journal of Bioinspiration & Biomimetics, 12(3):036006, 2017.

[4] H.V. Phan, T.N. Truong, and H.C. Park. Stable controlled flight of an insect-like tailless flapping-wing MAV. In Korean Society for Aeronautical and Space Sciences Fall Conference, 2016.

[5] H.V. Phan and H.C. Park. Remotely controlled flight of an insect-like tailless flapping-wing micro air vehicle. The 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 2015.

[6] S. Healy and T.A. Hurly. Hummingbirds. Current Biology, 16(11):R392-R393, 2006.

[7] R. Dudley. The biomechanics of insect flight: form, function, evolution. Princeton University Press, Princeton, NJ, 2002.

[8] M. Keennon, K. Klingebiel, H. Won, and A. Andriukov. Development of the Nano Hummingbirds: a tailless flapping wing micro air vehicle. 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2012.

[9] S.B. Fuller, E.F. Helbling, P. Chirattananon, and R.J. Wood. Using a MEMS gyroscope to stabilize the attitude of a fly-sized hovering robot. International Micro Air Vehicle Conference and Competition, 2014.

[10] S. Tijmons, M. Karasek, and G. de Croon. Attitude control system for a lightweight flapping wing MAV. Bioinspiration & Biomimetics, 13(5), 2018.

[11] D. Coleman, M. Benedict, V. Hrishikeshavan, and I. Chopra. Design, development and flight-testing of a robotic hummingbird. In American Helicopter Society 71st Annual Forum, 2015.

[12] F.Y. Hsiao, H.K. Hsu, C.L. Chen, L.J. Yang, and J.F. Shen. Using stereo vision to acquire the flight information of flapping-wing MAVs. Journal of Applied Science and Engineering, 15(3):213-226, 2012.

[13] J.L. Verboom, S. Tijmons, C.D. Wagter, B. Remes, R. Babuska, and G.C.H.E. de Croon. Attitude and altitude estimation and control on board flapping wing micro air vehicle. IEEE International Conference on Robotics and Automation (ICRA), 2015.

[14] G.M. Siouris. An engineering approach to optimal control and estimation theory, John Wiley & Sons, Inc, New York, NY, 1996.

[15] F.L. Lewis. Applied optimal control and estimation. Prentice Hall International, New Jersey, NJ, 1992.

[16] H.V. Phan, S. Aurecianus, T.S Kang, and H. C. Park. Attitude control mechanism in an insect-like tailless two-winged flying robot by simultaneous modulation of stroke plane and wing twist. International Micro Air Vehicle Conference and Competition, 2018.

[17] M. Pedley. Tilt sensing using a three-axis accelerometer, application note AN3461. Freescale Semiconductor, 2007.

[18] AN3182 application note, tilt measurement using a low-g 3-axis accelerometer. STMicroelectronics, 2014.

[19] V. Grygorenko. Sensing – magnetic compass with tilt compensation – AN2272. Cypress, 2014.

Aerodynamic investigation of the free flapping flight of a Saker falcon using a 3D multi-view reconstruction method

Martin Heinold[*], Christian J. Kähler
Universität der Bundeswehr München, Werner-Heisenberg-Weg-39, 85577 Neubiberg, Germany

ABSTRACT

This paper presents the process of reconstructing a three dimensional point cloud of a Saker falcon's geometry surface from five stereo camera pairs which are mounted on the top, bottom and side of a wind tunnel test section. To obtain three-dimensional point cloud information, corresponding surface point pairs are determined by rectifying camera images and automatically calculating sparse matching features. With the help of a Deep Flow Optical Flow algorithm (DFOF) a dense displacement field is calculated. Specific challenges associated with the 3D reconstruction of a Saker falcon in free flapping flight will be discussed in the paper. Furthermore the flapping flight data of the Saker falcon (*Falco cherrug*) at a mean chord based Reynolds number of about 250,000 will be presented in this paper. An aerodynamic analysis of the free flying falcon, including lift, drag, flight path and angle of attack, will be presented. The analysis shows that the movement of a complete stroke cycle and the structure of the wing surface of the bird can be reconstructed with sufficient accuracy for aerodynamic analysis. This implies that many aerodynamic aspects can be resolved that were not known before. Some examples are offered in the manuscript.

1 INTRODUCTION

The dream of conquering the sky by flying like a bird has been around for many centuries. Most notably Leonardo da Vinci wrote a manuscript in 1505 titled "Codex on the Flight of Birds" in which he details his observations of birds as well as plans and drawings of aircraft. Even though his flight tests failed due to insufficient muscle power, they have laid the foundation for following researchers like Otto Lilienthal or the brothers Wright and continuous research to this day. It is for this reason that Leonardo da Vinci is usually called founder of bionics, the link of biology and technology. Significant aims of the research is to improve the efficiency and optimize the aerodynamic design by learning from a biological model. However, it must be stated that the general believe that the biological selection process leads to optimal designs is wrong for a number of reasons. First, evolutionary processes are always based on specific materials, that must not be optimal for the task. Second, once a structure has evolved for any reason it is not possible to get rid of it in case it limits the performance. Therefore, third the evolutionary process leads most often to locally optimized structures but not to globally optimized designs. In view of flying animals we know birds, insects, bats, fish, flying foxes and ancient flyers. All can fly but non can be considered as a perfect flyer. Fourth, it is not possible to optimize quantities at the same time, that depend on more than one parameter i a non trivial manner. The flapping flight of birds is a reasonable propulsion method at Reynolds numbers about 250000. In a fluid with constant viscosity μ, these Reynolds numbers occur according to $Re = \rho \cdot u \cdot l / \mu$ when the length of the flight object l (which is often the chord of the wing) is sufficiently small or the flight speed u or rather the density of the fluid ρ is low enough. During a flapping cycle the bird wing carries out a strong deformation of the wings surface, which affects the aerodynamics.

To relate the flapping flight motion of the bird with the lift $L = \rho b u \Gamma$ and drag D the deformation of the wing needs to be determined with a contactless measurement method. This technology is able to resolve the flapping motion of the wing over a complete flapping period to measure the wing surface area, the wing span b and the continuous variation of the angle of attacks to determine the circulation Γ. Within the scope of a joint research project (DFG-SPP1207) Bachmann and Wagner [1], Friedl and Kähler [2] and Wolf and Konrath [3] investigated the gliding flight of barn owls (*Tyto alba*). The researchers evaluated and compared different experimental methods setups. Stationary and moving camera systems were examined. Moving imaging systems have shown advantages during the evaluation process of the data because the bird needs more time to pass the observation field, which allows for the capturing of more than one flapping cycle. On the other hand the background separation is rather complicated because the background is moving. Furthermore a moving camera frame can irritate the bird and leads to unnatural behavior. Stationary cameras simplify the masking of a moving object from a fixed background. Other 3D-measurement tech-

[*]Email address(es): martin.heinold@unibw.de

niques like Time-of-Flight cameras [4] allow for calculating distances by measuring light travel time. However usage of more than one system or background light can lead to disturbances with uncertainties of mm up to cm. Ponitz et.al. [5] recorded peregrine falcon (*Falco peregrinus*) during dive experiments along a dam wall. The trajectory of the falcon was calculated with the help of stereo reconstruction of a single marked point on the bird. Three more cameras took images of the falcon from different perspectives to understand the motion and the shape of the bird. Thereby it was possible to explain the diving flight, but the wings fit closely to the body during that maneuver to reduce drag. Only for the change of direction the wing was folded out. The flapping motion was studied by Carruthers et.al. [6] with six cameras and a Steppe Eagle (*Aquila nipalensis*). The research showed that the leading edge of the bird is important for the aerodynamic analysis. The relatively low recording rate of $25\,\mathrm{fps}$ demonstrates problems with a complex and fast motion like the flapping flight. In that case a flapping cycle lasts about $0.2 - 0.25\,\mathrm{s}$ and only $4 - 6$ images resolve the whole motion. Positions like the upper and lower dead center need to be precisely resolved over time because the curvature, the angle of attack and the wing shape change significantly. Additionally surface points were chosen by manually clicking 390 correspondence points and reconstruct these. The precision and the reproducibility seem to complicated. Therefore a system was developed that is able to automatically reconstruct the wing surface of a flapping bird with a sufficient recording rate. To obtain 3D point clouds from multiple camera images our method uses procedures from Computer Vision. After a volumetric calibration (mathematical relation between 2D image coordinates (x, y) and 3D real world coordinates (X, Y, Z)), correspondences between two or more cameras are obtained by automatic calculation of sparse feature points. With the help of a DFOF algorithm a dense disparity map is determined and triangulated to get 3D coordinates. Aerodynamic characteristics, like angle of attack variations, wing half span s, the flapping frequency f_{falcon} the falcon speed u_{falcon} as well as lift L are presented in this paper.

2 Measurement setup

The atmospheric wind tunnel of the Bundeswehr University Munich was used for the measurements of flapping flight of a Saker falcon (*Falco cherrug*). Two birds, a male ($720\,\mathrm{g}$, wingspan about $1\,\mathrm{m}$) and a female ($1040\,\mathrm{g}$, wingspan about $1.1\,\mathrm{m}$) were trained to fly through the wind tunnel against a uniform flow with velocities from $10\,\mathrm{m/s}$ to $18\,\mathrm{m/s}$. As a result of the headwind, the time of flight in the measuring area increases and more than one flapping period can be recorded to make a statistical analysis of the whole flapping cycle possible. The test section of the open loop wind tunnel has a cross-section of $1.85\,\mathrm{m} \times 1.85\,\mathrm{m}$ and measures $22\,\mathrm{m}$ in length. The facility can produce velocities between $2\,\mathrm{m/s}$ and $45\,\mathrm{m/s}$. The turbulence level in the test section was measured

Figure 1: Male Saker falcon with natural pronounced texture on the upper and lower body

$\mathrm{Tu} = \sqrt{\overline{u'^2}}/u_\infty = 0.5\%$. That imitates real world conditions in the lower atmosphere near the ground. The data was acquired simultaneously with 10 pco.dimax highspeed cameras from PCO. The camera sensor of these cameras generates full frame images with $2016\,\mathrm{px} \times 2016\,\mathrm{px}$ resolution, recorded with $1279\,\mathrm{fps}$. To resolve the motion of a wing up- and downstroke over time precisely, a frame rate for the recordings of $1000\,\mathrm{fps}$ was sufficient. The highspeed cameras with baselines of $800 - 1200\,\mathrm{mm}$ were equipped with Zeiss Distagon $35\,\mathrm{mm}$ lenses (f-number of 11) with a working distance between camera and measurement object of approximately $1\,\mathrm{m}$.

3 Camera calibration

To reconstruct a 3D point cloud with more than two cameras into the identical reference coordinate a camera pinhole model is reasonable. The camera matrices $\mathbf{P}_i$ of every camera consist of intrinsic ($\mathbf{A}$) and extrinsic parameters ($\mathbf{K}$). $\mathbf{A}$ contains the focal length of the objective, the physical size of each pixel on the camera sensor s (s_{x} and s_{y} for non-square pixels) and the principal point u_0 and v_0. To obtain the mathematical conversion from the three dimensional world coordinates $\mathbf{X} = (X, Y, Z)^{\mathrm{T}}$ to sensor coordinates $\mathbf{x} = (x, y)^{\mathrm{T}}$ and the relative rotation $\mathbf{R}$ by Euler angles and translation $\mathbf{t}$ of the camera, marked points on a two plane calibration target are detected (see Figure 2). The calibration pattern is moved rotatory and translational through the measurement domain. The reference coordinate system is setup at the first position of the pattern where the $X - Y$-plane is plane-parallel to the target and the Z-axis is perpendicular to it. To align the world coordinate system with the wind tunnel walls three reference points each on the top and bottom of the channel allow for transforming the coordinate system so that the X-axis is parallel to the streamwise direction. Z is from the bottom to the

Figure 2: Two plane calibration target (LaVision type 31): point to point distance $15\,\mathrm{mm}$, point diameter $3\,\mathrm{mm}$, plane height $3\,\mathrm{mm}$, plate thickness $14.8\,\mathrm{mm}$ [7]

top of the wind tunnel and Y is perpendicular to X and Z.

To transform a point from a 3D space onto the 2D camera coordinate system successive multiplication and addition is required. The reduction of complexity is achieved by using homogeneous coordinates. As a result the transformation is applied as a matrix multiplication.

$$\begin{aligned}\begin{pmatrix} x^* \\ y^* \\ 1 \end{pmatrix} &= \mathbf{K} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \\ &= \left[R_z \cdot R_y \cdot R_x \middle| \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \right] \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}\end{aligned} \tag{1}$$

$$\begin{aligned}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} &= \mathbf{A} \cdot \begin{pmatrix} x^* \\ y^* \\ 1 \end{pmatrix} \\ &= \begin{bmatrix} \frac{f}{s} & 0 & u_0 \\ 0 & \frac{-f}{s} & -v_0 + h \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x^* \\ y^* \\ 1 \end{pmatrix}\end{aligned} \tag{2}$$

Distortion on the basis of lens curvature, like radial and barrel distortion, affect the pinhole assumption of straight lines between 3D world points and the camera center. By calculating the undistorted pixel coordinates with the distortion coefficients this bias can be corrected. Due to the production quality of the Zeiss lenses distortions play only a minor role, despite everything high order polynomials correct the distorted pixel coordinates [8]. Basis for the stereo reconstruction is the projective geometry between two cameras. Corresponding point pairs $\mathbf{x} = (x, y)$ and $\mathbf{x}' = (x', y')$ from the calibration pattern comply with the fundamental equation.

$$\mathbf{x}'^T \cdot \mathbf{F} \cdot \mathbf{x} = 0 \tag{3}$$

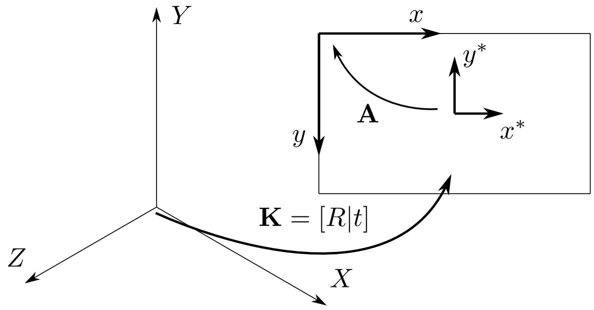


Figure 3: Translation t and rotation R of the world coordinate system ((X, Y, Z) in m) on to the camera sensor center ((x^*, y^*) in m). Afterwards recalculation of the camera coordinate system into px units ((x, y) in px)

$\mathbf{x}'^T \cdot \mathbf{F}$ and $\mathbf{F} \cdot \mathbf{x}$ describe respectively projected lines from one point $\mathbf{x}'$ or $\mathbf{x}$ to a line $\mathbf{l}'$ or l on the other camera sensor in each case. All lines intersect in one point, the epipole. To reduce the complexity of the correspondence search, the epipoles on both sensors are mapped to infinity by means of $\mathbf{T}_1$ and $\mathbf{T}_2$ [9]. This process is called rectification and is displayed In Figure 4. Lines, intersecting only at infinity, need to be horizontal. Hence the correspondence point search reduces to a one dimensional problem along the epipolar lines.

4 Dense disparity map calculation

The shift between one characteristic point in two images along the epipolar line is called disparity (see Figure 4). The displacement is inversely proportional to the depth of a point. In principle only the length of the baseline and the focal lengths of the cameras, both information from the camera calibration, are required to calculate the depth of a 3D world point as shown in equation (4) [10].

$$d = x - x' = \frac{Bf}{Z} \tag{4}$$

The determination of corresponding point pairs in camera setups with huge baselines leads to ambiguities and occlusions. To reduce the analysis effort of calculating dense displacement maps the first step is to detect sparse corresponding features.

The relevant part of the image, the flapping falcon, is segmented from the resting background by subtracting the images, preprocessed with a canny edge filter, and carry out a logical xor operation. Remaining noise can be removed with an area opening. Parts of the bird that overlap with edges from the background need to be filled by an erosion and a morphological fill function. The masks can be temporally smoothed with the help of Optical Flow. Five consecutive time steps are back projected to the initial time step and a median filter is applied (see Figure 5).

Brox et.al. [11] proposed a formulation of Optical flow that minimizes the energy function in equation (5)

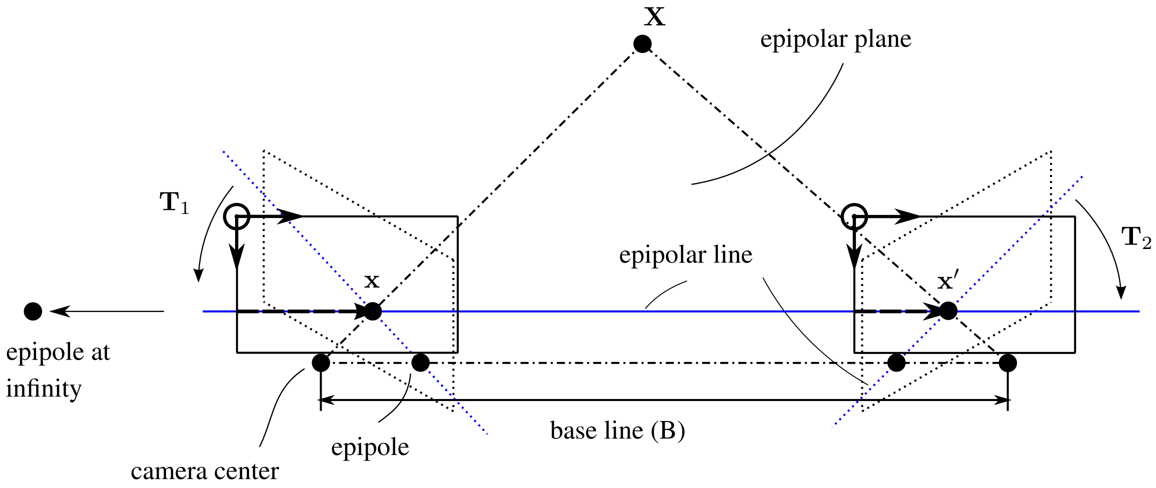


Figure 4: Rectification by means of epipolar geometry: a 3D world point $\mathbf{X}$ is displayed on two camera sensors at the points $\mathbf{x}$ and $\mathbf{x}'$. Epipoles calculated with the help of corresponding calibration points are mapped to infinity by multiplication of transformation matrices $\mathbf{T}_1$ and $\mathbf{T}_2$

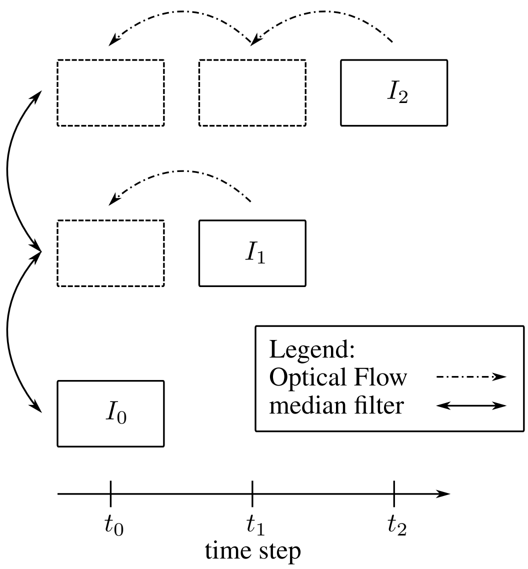


Figure 5: Schematically presentation of temporal smoothing with 3 time steps

$$
\begin{aligned}
E(\mathbf{w}) = &\int_\Omega \Psi(|I(\mathbf{x}+\mathbf{w}) - I(\mathbf{x})|^2)\mathrm{d}\mathbf{x} \\
&+ \gamma \int_\Omega \Psi(|\nabla I(\mathbf{x}+\mathbf{w}) - \nabla I(\mathbf{x})|^2)\mathrm{d}\mathbf{x} \\
&+ \alpha \int_\Omega \Psi(|\nabla u|^2 + |\nabla v|^2)\mathrm{d}\mathbf{x}
\end{aligned}
\tag{5}
$$

with the following comments:

$$\mathbf{w} = (u, v, 1)^{\mathrm{T}} \tag{6}$$

$$\Psi(s^2) = \sqrt{s^2 + \epsilon^2} \text{ with } \epsilon = 0.001 \tag{7}$$

The robust expression with the help of l_2 or Euclidean norm is presented in equation (7). Equation (5) contains a constant intensity term, a constant gradient term along edges and a velocity field smoothness term. The parameters for regularization α, the weight γ and the scaling factor of the Gaussian pyramid were determined by Heinold and Kähler [12].

In Figure 6 the falcon image and the resulting mask is shown. On the leading edge of the right wing a streak is visible that over segments the falcon. Due to different lighting situations during calibration and the recording, areas in the background can get a higher weight during edge detection and remain in the image after segmentation. On the basis of the stripe that is connected to the birds mask, the area open method does not remove this part. To avoid losing information from the wing, over segmentation is preferred compared to under segmentation where areas of the bird are missing.

Only within the masks corresponding points are searched to reduce the computational expense. Furthermore a wing flap consist of about 250 to 300 images but only every 10th

Figure 6: Foreground / Background segmentation of a Saker falcon by background subtraction: left image – rectified falcon image from top view, right image – mask of the falcon

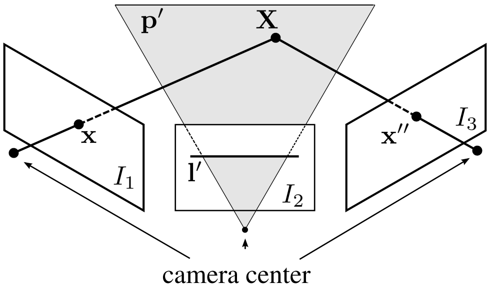


Figure 7: Transfer from one point $\mathbf{x}$ via plane $\mathbf{p}'$ to point $\mathbf{x}''$ (following [19, p. 382])

images is used to detect sparse feature points. To reduce noise and enhance high frequency signals in the image, at first the stereo images are subtracted of the sliding minimum (window size $9\,\mathrm{px}\times 9\,\mathrm{px}$). Possible prominent features are corners, that can be located with a minimum eigenvalue algorithm [13]. All corner points in one camera are normalized cross correlated with stripes of the second camera image on the same epipolar line [14]. Four different window size ($12\,\mathrm{px}\times 12\,\mathrm{px}$, $24\,\mathrm{px}\times 24\,\mathrm{px}$, $48\,\mathrm{px}\times 48\,\mathrm{px}$ and $64\,\mathrm{px}\times 64\,\mathrm{px}$) are used to detect feature in various sizes. Only correlation maxima higher than 1.5 times the correlation value of the second highest peak are considered matches [15]. To locate the position with subpixel accuracy a gaussian fit is implemented. The process is repeated with corners from the second camera and stripes in the first camera image. To increase the number of matches an improved SIFT-algorithm is applied [16, 17, 18]. The corresponding point pairs are searched for outliers by using a RANSAC algorithm for a homography [19].

With the help of Optical Flow the matching point pairs are distributed back and forth in the timeseries (see Figure 9). To prevent matches to be wrong, point pairs from one time step to another are cross correlated. The shift between two time steps is significantly lower than between two different, rectified cameras. Therefore the selection process of matches simplifies to choosing matches with normalized correlation values higher than 0.8.

To increase the amount of point correspondences more than two cameras can be used. With the help of a Trifocal ten-

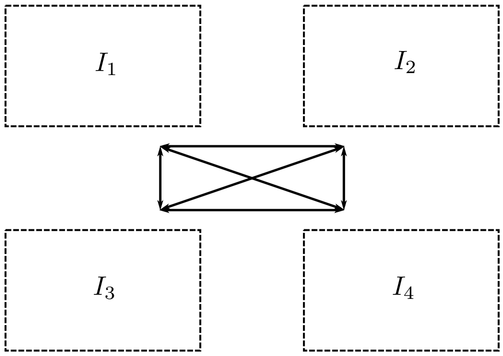


Figure 8: For every set of three cameras with overlapping point of view a Trifocal tensor can be set up

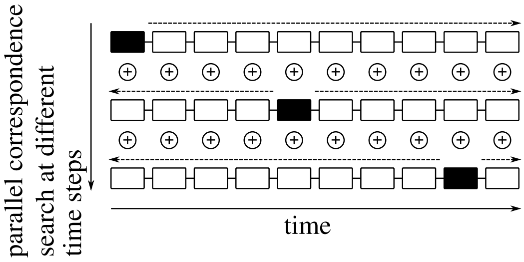


Figure 9: Diagram of the parallel search for corresponding points and spread out with Optical flow (dotted arrow) to time steps before and after, schematically depicted for every 4th time step (black box). Finally summation of all correspondences for one time step

sor T_f it is possible to perform a point transformation from two cameras to a third one [19] (see Figure 7). In Figure 8 it is shown that every two cameras from a set of cameras with overlapping point of view can transfer corresponding points $\mathbf{x}$ and $\mathbf{x}'$ to a third camera $\mathbf{x}''$. To calculate the coordinate in the third view a line $\mathbf{l}'$ is selected that passes through $\mathbf{x}'$ and spans a plane $\mathbf{p}'$. With that the position $\mathbf{x}''$ is determined in tensor notation (compare [19, p. 376]) by:

$$x''^{k} = x^{i} l'_{j} T_{i}^{jk} \tag{8}$$

The search for corresponding point pairs can be parallelized because the individual calculations every 10th time step are independent from each other. Therefore the distribution to more workers reduces the evaluation time about the factor of the number of workers. The small difference occurs by virtue of more data copying processes there and back to every single worker. During the analysis a PC with an Intel Core i7-4930K processor (6 cores with $3.4\,\mathrm{GHz}$ each), $32\,\mathrm{GB}$ RAM and a Nvidia GeForce GTX 660 (960 Shaders, $993\,\mathrm{MHz}$ GPU Clock, $2\,\mathrm{GB}$ RAM) was used. The evaluation was done with MATLAB R2017a and OpenCV 3.2.0.

The evaluation showed that two different formulations of

Figure 10: Left image: spatial smoothed disparity map, right image: spatial and temporally smoothed disparity map – gray values from dark gray to white indicate higher disparity and lower depth value – mask is applied to the picture (black background)

the Optical Flow are necessary for the computation of a dense disparity field. For image sequences with small shifts between consecutive images equation (5) from Brox et.al. [11] shows strong results. On the other hand an extension of the formulation is required for the computation of correspondences between different cameras. With the input of sparse corresponding point pairs a Deep Flow Optical Flow algorithm is able to calculate a dense disparity map [20].

$$\begin{aligned} E(\mathbf{w}) = & \int_{\Omega} E_{\text{Data}} \mathrm{d}\mathbf{x} \\ & + \alpha \int_{\Omega} E_{\text{Smoothnes}} \mathrm{d}\mathbf{x} + \chi \int_{\Omega} E_{\text{Matching}} \mathrm{d}\mathbf{x} \end{aligned} \tag{9}$$

The third term in equation (9) includes information of the sparse correspondences. Furthermore the variable χ determines the weight of the sparse matches as shown in Stoll et.al. [21].

The Deep Flow algorithm spatially smooths the disparity map. To close holes in the displacement field and remove outlier disparities a temporal smoothing method, as explained on page 3, is applied to the disparity map.

5 Triangulation of the pointcloud

Every value in the disparity map represents a displacement from one image to another. These corresponding points can be triangulated into 3D space with the help of the camera matrices $\mathbf{P}_i$ and the fundamental matrix $\mathbf{F}$. Using the optimal triangulation method proposed in [19], three dimensional coordinates (X, Y, Z) are calculated and thereby the sum of squared distances is minimized in accordance to the fundamental equation (3).

To reduce the amount of remaining outliers to consecutive steps are used. At first the triangulated world points $\mathbf{X}$ are back projected on the left camera sensor $\hat{\mathbf{x}}$.

$$\hat{\mathbf{x}} = \mathbf{P}_1 \cdot \mathbf{X} \tag{10}$$

Only points which are located inside the left image mask are kept. The second step outliers are detected by Sparse Outlier Removal algorithm [22]. The foundation of the program

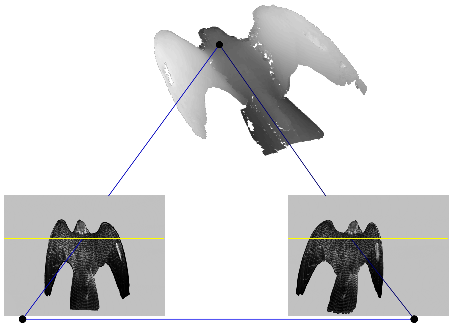


Figure 11: 3D point cloud reconstruction

is a nearest neighbor search around every point within a specified amount of points (during the evaluation 200 was chosen). The point distribution is examined and points outside a certain allocation are removed.

6 Uncertainty analysis

It is difficult to define an uncertainty for a stereo reconstruction algorithm. One of the most used kind of uncertainty is the re-projection error [23]. To determine the error there is a similar approach like in section 5. The world coordinates are back projected on the camera sensor by multiplying the camera matrices $\mathbf{P}_i$ onto the 3D point (see equation (10)). With the resulting n point positions $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ the geometrical distance to the original points $\mathbf{x}$ and $\mathbf{x}'$ is calculated.

$$\text{Err} = \sum_{i}^{n} d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \tag{11}$$

During the analysis an error Err of $0.3965\,\mathrm{px}$ was measured.

To make further statements about the uncertainty in an experimental setup prior to the measurement campaign with the Saker falcon a well-defined plate with a random dot pattern on it was moved by a traversing unit. The plate had a thickness of $8\,\mathrm{mm}$ and it was moved along the Z-axis of the coordinate system through the measurement domain in $25\,\mathrm{mm}$ steps.

The plate was reconstructed for 9 consecutive steps and the mean displacement between the moves is $24.48\,\mathrm{mm}$. The mean relative difference to the real movement is therefore calculated to $0.52\,\mathrm{mm}$. Two cameras reconstructed the upper side of the plate and two cameras the lower side. Hence the thickness of the aluminum plate was calculated and the absolute mean deviation was $0.19\,\mathrm{mm}$ from the real size.

7 Aerodynamic analysis

During the measurement campaign over 50 flights of two falcons (one female, one male) in the Atmospheric Windtunnel Munich at different incoming flow velocities were performed. This paper focuses on one flight of the female Saker falcon at $u_\infty = 18\,\mathrm{m/s}$ wind speed. During the flight through

Figure 12: Well-define aluminum plate with random dot pattern, thickness $8\,\mathrm{mm}$ – moved by traversing unit along Z-axis in $25\,\mathrm{mm}$ steps

730 time steps were recorded and are evaluated for the analysis. The coordinate system is an aerodynamic coordinate system with the following orientation: X is in the inflow direction, Y along the wingspan of the right wing and Z depicts the height. The subscript $b, t = 0$ means a fixation of the coordinate system in the first time step of the evaluation at a prominent point of the back of the bird $(X_{\mathrm{b,t=0}}, Y_{\mathrm{b,t=0}}, Z_{\mathrm{b,t=0}})$. Furthermore to simplify the discussion a non-dimensional coordinate along the Y-axis is defined by dividing the Y-position with the current half-span s in every time step. To compare different flaps with each other a phase angle is introduced. $0°$ is the highest body position, $180°$ the lowest one (compare Figure 13).

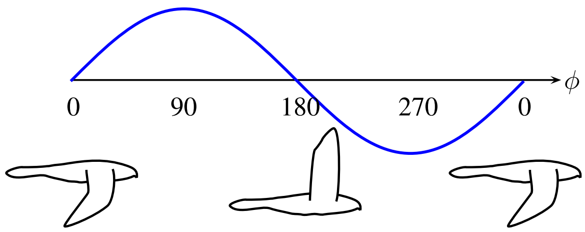


Figure 13: Definition of phase angle ϕ.

$$\zeta(Y, \phi) = \frac{Y(\phi)}{s(\phi)} \tag{12}$$

Figure 14 shows the variation of the wing half-span over the phase angle. During the up- and downstroke of the third flap the outermost feathers of the wing are out of the image. Additionally the lighting of the hand wing is faint. Therefore the local maximum of the half span is underestimated in comparison to the first two flaps which were almost perfectly

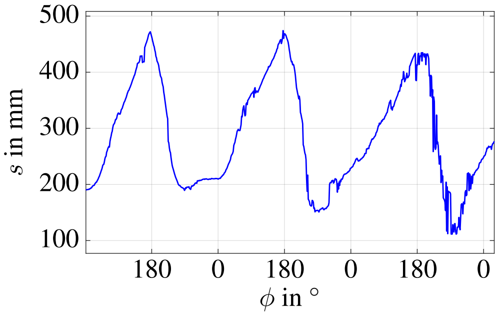
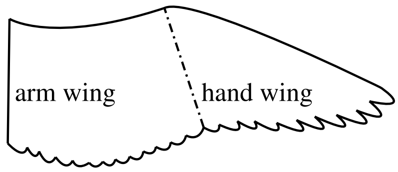
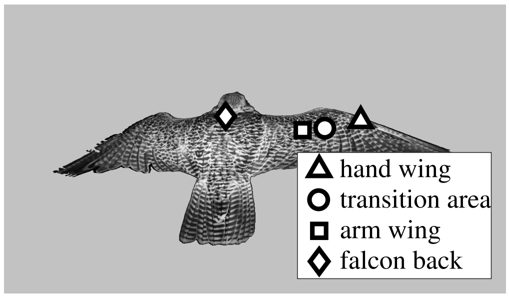


Figure 14: Maximum wing half span $s_{\max}$ plotted over phase angle ϕ.

recorded. The analysis of the first two flapping cycles shows the high reproducibility of the flapping. This demonstrates the abilities of the falcon and our developed measurement technique. Read from Figure 14 the global wing half span is equal to $s_{\max} = 474\,\mathrm{mm}$.



(a)



(b)

Figure 15: (a): Diagram of a bird's wing (following [24, p. 30]), (b): Points viewed on the wing (arm wing – square, hand wing – triangle, transitional area – circle) and back of the bird (diamond).

The analysis of the bird aerodynamics can be quantitatively done by looking at the flight patch of the falcon. In Figure 15 four specific points on the top of the bird are regarded. One point in the middle of the falcon body and three different positions on the right wing. These are divided into one point on the arm wing ($\zeta = 0.308$), one on the hand wing ($\zeta = 0.539$) and the last one in the transitional area

($\zeta = 0.397$). In Figure 15 (a) a schematic representation of a falcon wing is shown. From this image it is recognizable that the structure of the wing allows for the variation of the leading edge bearing ψ during a flapping cycle. The selection of a point on the arm wing and one on the hand wing is relevant on the basis of two completely different angles ψ with regard to the Y axis. This characteristic produces various motions on the different wing positions what leads to different aerodynamic performance.

During a flapping cycle the wing lifts up followed by a downstroke. To understand the differences of a falcon flap in comparison to a fixed flapping wing the relative differences between the three wing positions and the movement of the body is shown in Figure 18. In Z direction all three points on the wing follow a sinusoidal motion and the amplitude increases towards further outlying wing positions (see Figure 18 (c)). This behavior would be similar to the mentioned fixed flapping wing. In Figure 18 (a) and Figure 18 (b) the large displacement of the hand wing and the relative small movement of the arm wing are presented. Figure 18 (b) shows a similar course of all three curves. Only in the range of extrema ($\phi \approx 190°$) does the amplitude increase significantly in the span direction. This can be explained above all by the high dynamics of the arm wing. For clarity the motion is depicted from at four different phase angles from a top view camera in Figure 17.

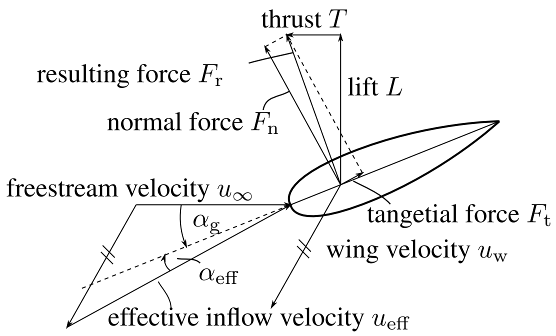


Figure 16: Lift and thrust generation of a moving wing on the basis of different angles of attack. Geometrical AOA α_g for a steady wing. For analysis, the wing is approached at the effective AOA α_eff. [25]

In Figure 19 the motion of the point on the back of the bird is depicted over the phase angle ϕ. Because a 3D representation is to complicated the three-dimensional flight path is shown in three different lines of sight ($X-\phi, Y-\phi, Z-\phi$). The movement in Figure $Z-\phi$ presents a sinusoidal form and a downward gliding motion. To conserve angular momentum the natural reaction to compensate the wing motion is the up and down movement of the body. As a result the maximum position of the bird body corresponds nearly to the local minimum of the wing and vice versa. During a upstroke of the wing the body is lowered and the other way around. There is a small lag of the wing motion in comparison to the body. The downward movement during the first flapping cycle can be explained by the fact that the falcon must first determine its target after the start and then moves to the corresponding position and height. The Figure $Y - \phi$ shows that the falcon does not fly precisely against the incoming flow direction. This observation can be explained by the target of the falcon. To start a flight the bird is attracted by a falconer with some food (a common quail or a parts of a rat). As a predator the falcon focuses on the object and changes the flight path during the measurement to the right and downwards. Due to the high steady incoming flow velocity the flight distance in X direction is small. In nature without wind influence the real flight distance would be the measured distance divided by the falcon speed $u_\mathrm{falcon} = 1.36\,\mathrm{m/s}$ and multiplied with the total velocity $u_\mathrm{total} = u_\mathrm{falcon} + u_\infty = 19.36\,\mathrm{m/s}$. In this case the bird would have traveled roughly $13.5\,\mathrm{m}$ during the measurement. Combined with the downward motion in Z-direction during the flapping cycles the gliding angle is determined by equation (13).

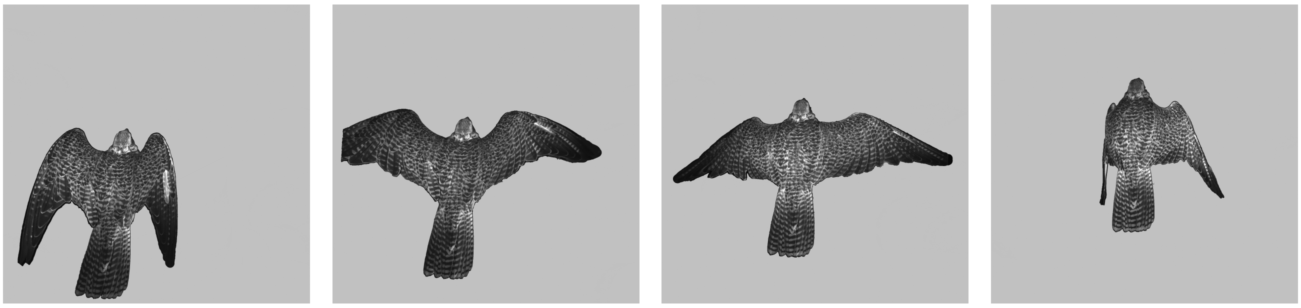
$$\gamma = \frac{\Delta Z}{\Delta X_\mathrm{total}} = -0.887° \tag{13}$$

In addition, the flapping frequency is determined with the aid of Figure 20 (a). For this purpose the highest local peaks of the periodic motion are detected to determine the frequency of $f_\mathrm{falcon} = 4.35\,\mathrm{Hz}$.

Figure 20 shows the movement of the prominent wing positions. Close to the falcon body the curve is smooth and monotonous. Rapid changes in the direction of movement on the arm wing lead to complex curves (see Figure 20 (c) and (b) at approximately $\phi \approx 0°$). Due to manoeuvring (slight right turn and descent), there is no complete symmetry between the wing flap (compare Figure 20 (c) arm wing motion at $\phi \approx 270°$).

$$L = c_\mathrm{L} \cdot \rho/2 \cdot u_\mathrm{total}^2 \cdot S \tag{14}$$

For a detailed aerodynamic evaluation the consideration of the wing profiles during the flapping cycle is necessary. Therefore, in the following section the wing cross-section at $\zeta = 0.55$ is examined. The wing profiles shown are taken from the second flapping cycle. In Figure 17 the pictures (a) to (d) show an upstroke of the wing, over the turning point and a downstroke (the highest point is reached at phase angle $\phi = 120°$). The evaluation of the profiles is limited to 70% of the total stroke due to the increasing uncertainty near the lower turning point as well as occlusions of the wing. The influence of aerodynamics on the bird is rather marginal at this time, since no lift is generated. Based on equation (14), the lift force is linked to the wing area. As this becomes minimal near the lower turning point, there is also only a minimal lift at this point. The airfoil cross-sections in Figure 21 are taken in a period of time between the pictures 17 (b) and (d).



(a) (b) (c) (d)

Figure 17: Snapshots from the flapping cycle of a Saker falcon ($\Delta t = 50\,\mathrm{ms}$). [25]

Based on these geometries Xfoil simulations were performed to estimate the aerodynamic properties of the profiles [26]. The free stream turbulence intensity of the Atmospheric Windtunnel Munich was measured to $Tu = 0.5\,\%$ at the used inflow velocities by Herbst et.al. [27]. To forecast the transition of 2D boundary layers in incompressible flows with the help of linear stability theory Van Ingen [28] developed the e^N method. Xfoil uses this method and calculates the N factor as shown in the following equation:

$$N = -8.43 - 2.4 \cdot \ln\left(\frac{Tu}{100\,\%}\right) \tag{15}$$

For further examination N is calculated as 4.2. It should be noted that the surface roughness of birds is higher than that of modern aircraft wings (compare Reichardt [29]). For this reason, lower N values of up to 1 would be possible to take the influence of surface roughness into account [26]. However, here we are interested in the sensitivity of aerodynamic performance to support aerodynamic analysis rather than absolute values.

The analysis of thickness and camber enables the calculation of geometric parameters. Figures 22 and 23 show positions and aerodynamic values of the profile in relation to thickness and curvature. In this phase of the stroke cycle, the position of the maximum thickness moves relative to the leading edge and the maximum thickness itself decreases significantly by about 50% before a further increase becomes apparent. Before phase angle $175°$ the average position X_{th}/c is 0.28. After the upper turning point, a position of about 0.14 is set. Based on the potential theory, the thickness distribution is responsible for the speed around the profile. An earlier position of the maximum thickness th/c indicates a greater curvature of the profile at the leading edge and thus a higher acceleration.

The maximum curvature f/c varies only slightly over the phase angle. However, the position of the maximum curvature X_{f}/c seems to move further back to the trailing edge until the upstroke becomes a downstroke. Then the value returns to a position of about 0.45.

Figure 24 shows the change in attack angles α_{g} and α_{eff} during the observed time period. During this stage of the flap, the effective AOA starts at high negative numbers, reaches $0°$ at the upper turning point and then rises further. These findings can be explained by the complex motion of the falcon. During a flapping cycle, not only lift has to be generated. The falcon also needs thrust to travel forward. The geometric AOA α_{g} of a constant inflow (negative for the downstroke, about $0°$ at upstroke) must be corrected by the movement of the wing. The effective AOA α_{eff} is shown in figure 24 (b). Therefore, a negative geometric angle of attack allows a vectorial decomposition of the attacking forces into lift and a horizontal part, thrust, as shown in figure 16. Even if the geometry of the profile changes during the stroke cycle, Figure 25 (b) shows a linear slope between $\alpha = -8°$ and $10°$ and the lift coefficient c_{l}. This allows to calculate a zero lift angle (α_0) of about $-3.2°$ and a lift coefficient at AOA $0°$ of $c_{\mathrm{l},0} = 0.4$.

The Reynolds number of a fixed wing under constant flow conditions (free flow velocity and viscosity) is calculated only once. Due to the large deformation and motion of the falcon wing, the Reynolds number changes at every phase angle. Nevertheless, the inflow conditions are almost uniform. With the definition of the Reynolds number:

$$Re = \frac{u \cdot c}{\nu} \tag{16}$$

Figure 24 can also be considered a variation of the chord length c. Before phase angle $175°$, the arm wing is near the body. Therefore, the chord length c at a relative position $\zeta = 0.55$ is so high at an upstroke, and rather steady in the process of the downstroke. Pulling up the wing during upstroke reduces drag and the profile generates downforce. As a result, the lift-to-drag coefficient $c_{\mathrm{l}}/c_{\mathrm{d}}$ in Figure 25 is slightly negative during this phase of the flap. The results during the downstroke show the efficiency of the natural propulsion system. Even with large wing motions and a upstroke phase, a lift coefficient of up to 50 can be achieved, which is a considerable value.

8 Conclusion

This paper presented a method to reconstruct a point cloud of the natural texture of a real Saker falcon flying in a wind-tunnel from the synchronized recordings of multiple cameras. To calculate the wing surface from both sides over a full flapping cycle 10 highspeed cameras were installed around the test-section of an atmospheric wind tunnel and recorded a flying bird in stiff headwind to increase the recording time. The system was calibrated with a camera pinhole model and lens distortions were corrected. Sparse correspondences were found with a correlation based method and a SIFT algorithm. The complexity was reduced by masking the falcon from the stationary background. The sparse point pairs allowed for usage of a Deep Flow Optical Flow algorithm. To increase the amount of correspondences a multi view method, the Trifocal tensor, was used. Most of the program allow for the parallelization on multiple workers to reduce the computational time. To obtain the 3D point cloud a optimal triangulation method is performed that minimizes the geometrical distance error. Finally the back projection error was measured and by means of a well-defined plate the uncertainty along the Z-axis was presented. Overall the method allows for the calculation of the wing surface of the bird over a time span. It has been shown that the resulting surface information can be used to analyze the aerodynamic properties like wing span, chord length, angle of attack and the variation over a flapping period without influencing the bird. This makes the qualitative aerodynamic analysis of the free flying birds possible and allows to deepen the locomotion of birds. The results show the high dynamic fluctuations of the aerodynamic properties during the flapping cycles. In addition, the analysis of a profile cross-section revealed a new perspective on what happens during up- and downstroke. Qualitative facts, such as the negative angle of attack at the stroke, made it possible to describe the complex movement of the falcon. The analysis shows the potential of the developed measuring system and the advantage of wind tunnel investigations for aerodynamic investigations of birds as other flying animals.

References

[1] Thomas Bachmann and Hermann Wagner. The three-dimensional shape of serrations at barn owl wings: towards a typical natural serration as a role model for biomimetic applications. *Journal of Anatomy*, 219:192–202, 2011.

[2] Alexander Friedl and Christian J. Kähler. Measuring and Analyzing the Birds Flight. In *STAB Fach-Symposium zur Strömungsmechanik*, 2010.

[3] Thomas Wolf and Robert Konrath. Avian wing geometry and kinematics of a free-flying barn owl in flapping flight. *Experimental Fluids*, 56:1–23, 2015.

[4] Larry Li. Time-of-flight camera–an introduction. *Technical White Paper, May*, 2014.

[5] Benjamin Ponitz, Anke Schmitz, Dominik Fischer, Horst Bleckmann, and Christoph Brücker. Diving-flight aerodynamics of a peregrine falcon (falco peregrinus). *PLoS ONE*, 9(2):1–13, 02 2014.

[6] Anna C. Carruthers, Simon M. Walker, Adrian L.R. Thomas, and Graham K. Taylor. Aerodynamics of aerofoil sections measured on a free-flying bird. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 224(8):855–864, 2010.

[7] LaVision. Personal correspondence with employee. 2015. Munich.

[8] OpenCV. *OpenCV 2.4.9 documentation*, 2016.

[9] Andrea Fusiello, Emanuele Trucco, and Alessandro Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12:16–22, 2000.

[10] OpenCV. *OpenCV 3.0 documentation*, 2017.

[11] Thomas Brox, Andres Bruhn, Nils Papenberg, and Joachim Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Warping. *Lecture Notes in Computer Science*, 3024:25–36, 2004.

[12] Martin Heinold and Christian J. Kähler. *Image and Video Technology: 7th Pacific-Rim Symposium, PSIVT 2015, Auckland, New Zealand, November 25-27, 2015, Revised Selected Papers*, chapter Wing-Surface Reconstruction of a Lanner-Falcon in Free Flapping Flight with Multiple Cameras, pages 392–403. Springer International Publishing, Cham, 2016.

[13] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[14] John P. Lewis. Fast normalized cross-correlation. *Vision interface*, 10(1):120–123, 1995.

[15] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1992.

[16] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[17] Relja Arandjelovic and Andrew Zisserman. Three Things Everyone Should Know to Improve Object Retrieval. In *Proceedings of the 2012 IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 2911–2918, Washington, DC, USA, 2012. IEEE Computer Society.

[18] Andrea Vedaldi. An open implementation of the SIFT detector and descriptor. Technical report, University of California at Los Angeles, 2007.

[19] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in computer vision*. Cambridge University Press, second edition, 2004.

[20] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching, 2013.

[21] Michael Stoll, Sebastian Volz, and Andrés Bruhn. Adaptive Integration of Feature Matches into Variational Optical Flow Methods, 2013.

[22] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[23] Simon M. Walker, Adrian L.R. Thomas, and Graham K. Taylor. Photogrammetric reconstruction of high-resolution surface topographies and deformable wing kinematics of tethered locusts and free-flying hoverflies. *Journal of The Royal Society Interface*, 6(33):351–366, 2009.

[24] Werner Nachtigall. *Warum die Vögel fliegen*. Rasch und Röhring, 1985.

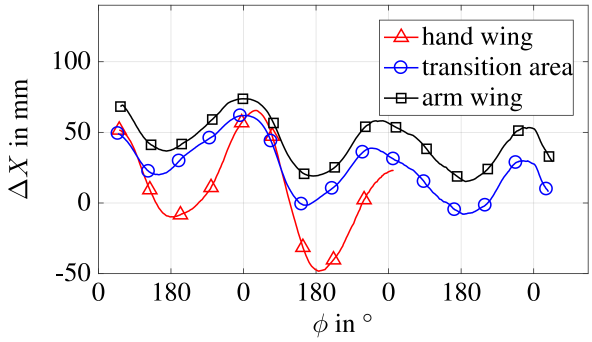
[25] Martin Heinold and Christian J. Kähler. Aerodynamic investigation of the free flapping flight of a saker falcon with the help of 3d multi-view reconstruction method. In *2018 Applied Aerodynamics Conference*, page 3173, 2018.

[26] Mark Drela and Harold Youngren. Xfoil 6.9. *User Guide. MIT*, 2001.

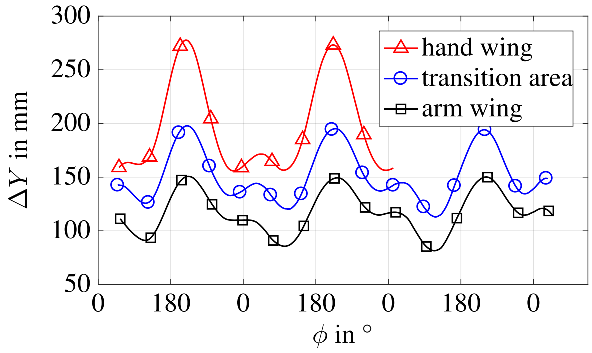
[27] Sebastian L. Herbst, Christian J. Kähler, and Rainer Hain. Sd7003 airfoil in large-scale free stream low-turbulence. In *47th AIAA Fluid Dyn. Conf.*, Denver, USA, 2017.

[28] Jan Van Ingen. The e^N method for transition prediction. Historical review of work at TU Delft. In *38th Fluid Dynamics Conference and Exhibit*, page 3830, 2008.

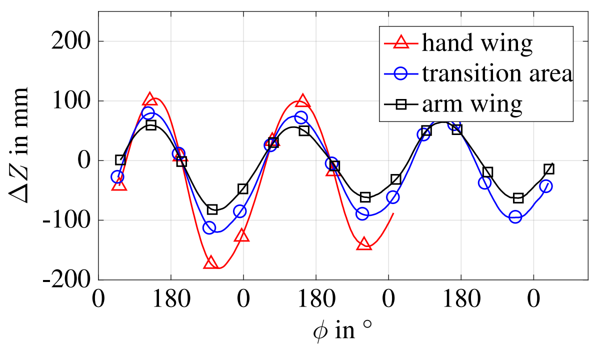
[29] Werner Reichardt. Verhaltensstudie der musterinduzierten Flugorientierung an der Fliege Musca domestica. In *Verhaltensstudie der musterinduzierten Flugorientierung an der Fliege Musca domestica. Biophysik des Tierflugs*, pages 7–72. Springer, 1974.



(a)

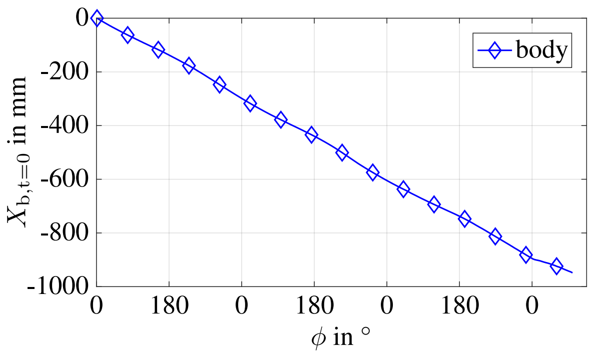


(b)

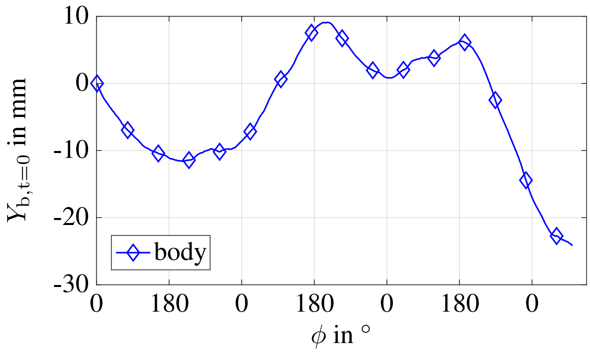


(c)

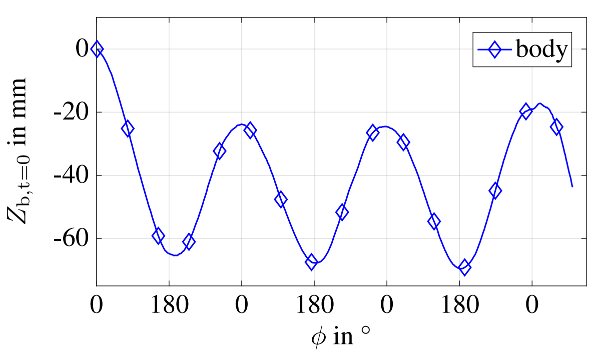
Figure 18: Distance of the three selected points on the wing to the point on the back of the bird displayed over the phase angle. Each plot shows a coordinate direction. (a): distance in X_b direction, (b): distance in Y_b direction, (c): distance in Z_b direction



(a)

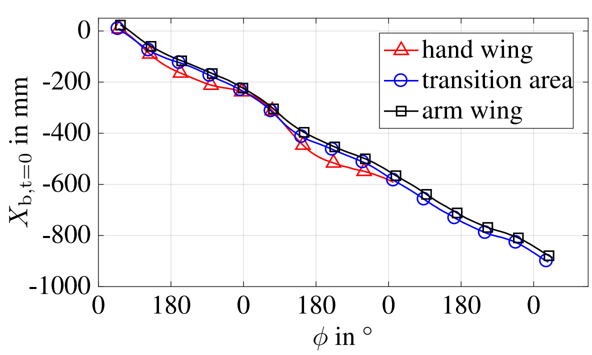


(b)

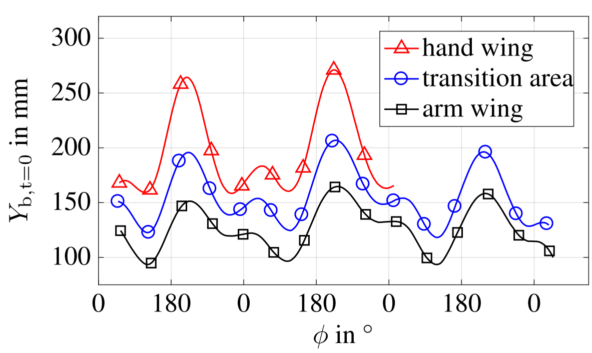


(c)

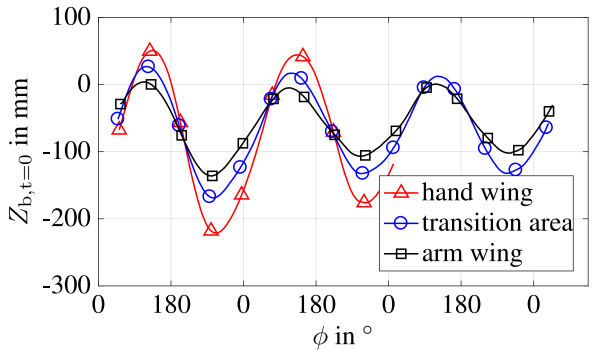
Figure 19: Movement of the body relative to a coordinate system on the back of the bird at the first time of evaluation t_0 applied over the phase angle. (a): motion in $X_{\mathrm{b,t=0}}$ direction, (b): motion in$Y_{\mathrm{b,t=0}}$ direction, (c): motion in $Z_{\mathrm{b,t=0}}$ direction



(a)

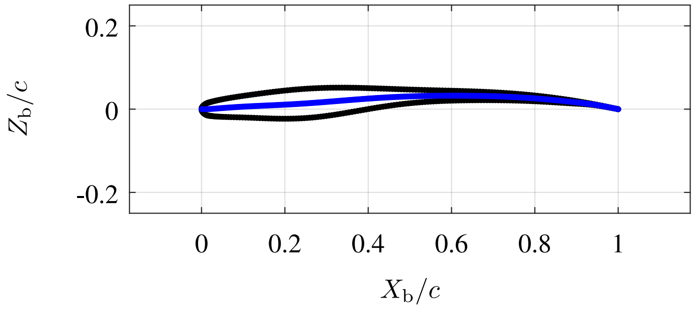


(b)

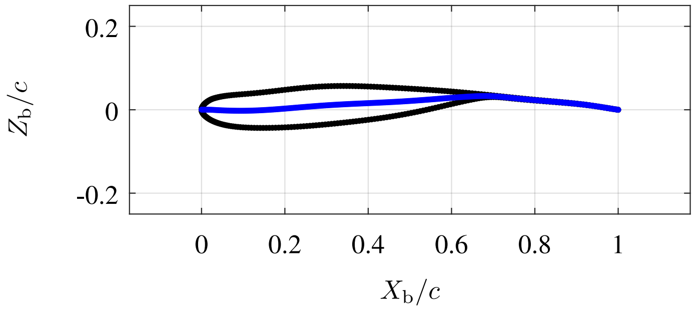


(c)

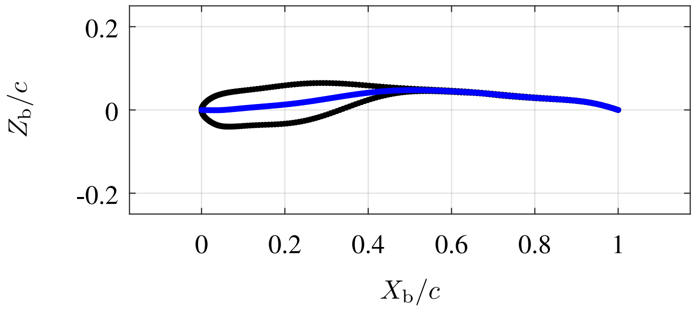
Figure 20: Movement of the Saker falcon wing relative to a coordinate system on the back of the bird at the first time of evaluation t_0 applied over the phase angle. (a): motion in $X_{\mathrm{b,t=0}}$ direction, (b): motion in$Y_{\mathrm{b,t=0}}$ direction, (c): motion in $Y_{\mathrm{b,t=0}} - Z_{\mathrm{b,t=0}}$ direction



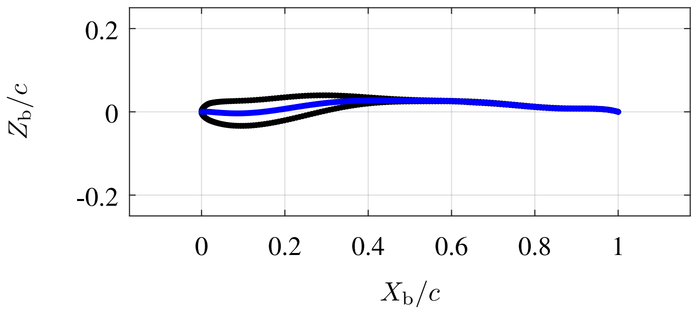
(a)



(b)

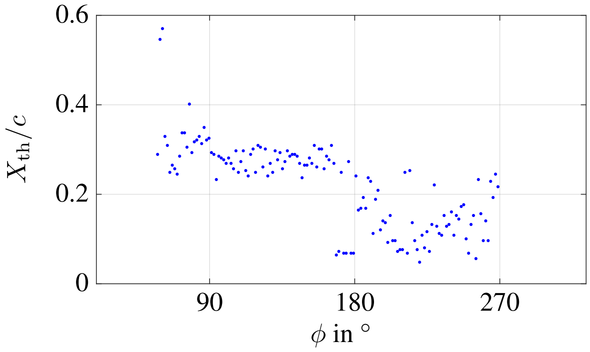


(c)

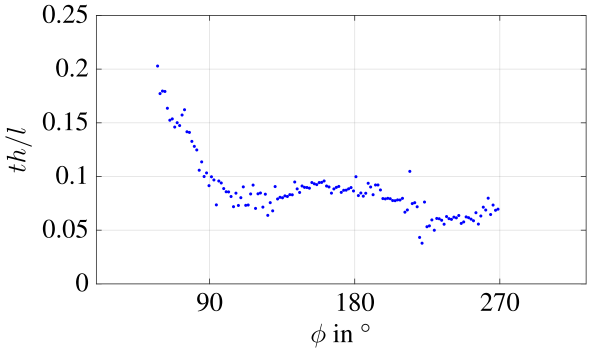


(d)

Figure 21: Wing cross section at $\zeta = 0.55$ on the right wing. $\Delta t = 25\,\mathrm{ms}$ between consecutive Figures.

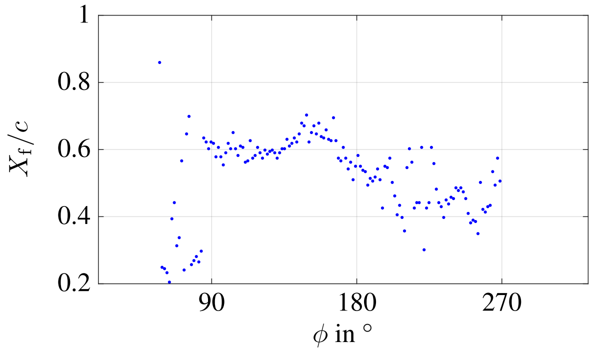


(a)

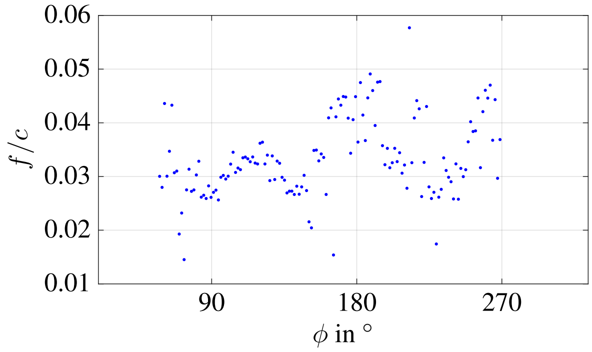


(b)

Figure 22: Distribution of maximum thickness values (th/c) and positions (X_{th}/c).



(a)



(b)

Figure 23: Distribution of maximum camber values (f/c) and positions (X_{f}/c).

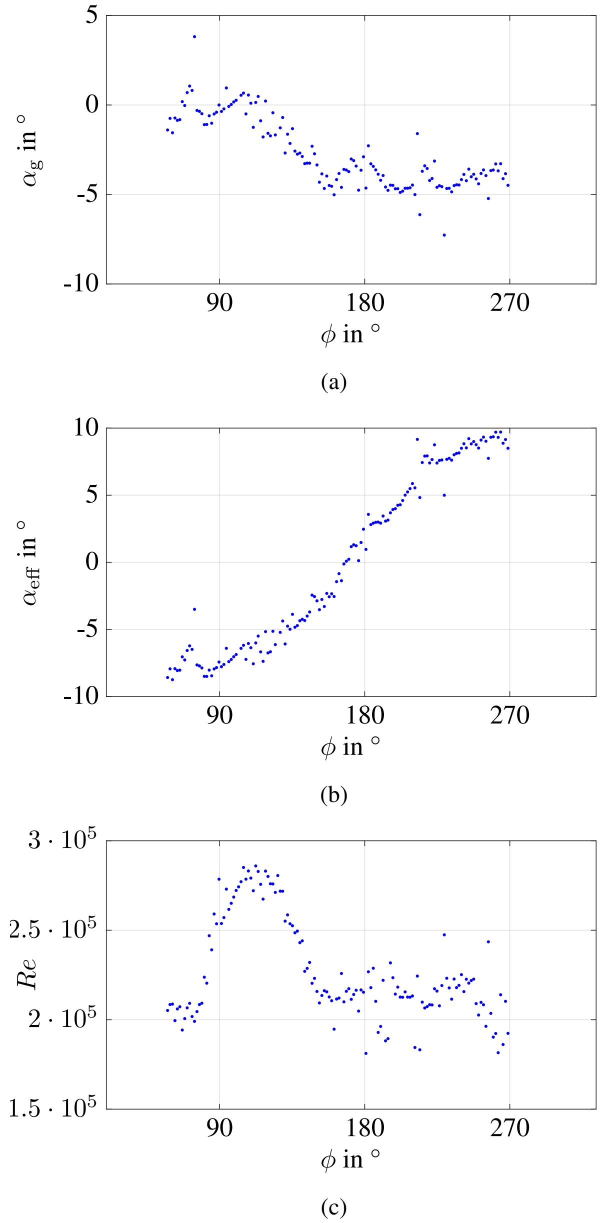


Figure 24: Angle of attack variation α_g and α_{eff} as well as variation of the Reynolds number Re over phase angle.

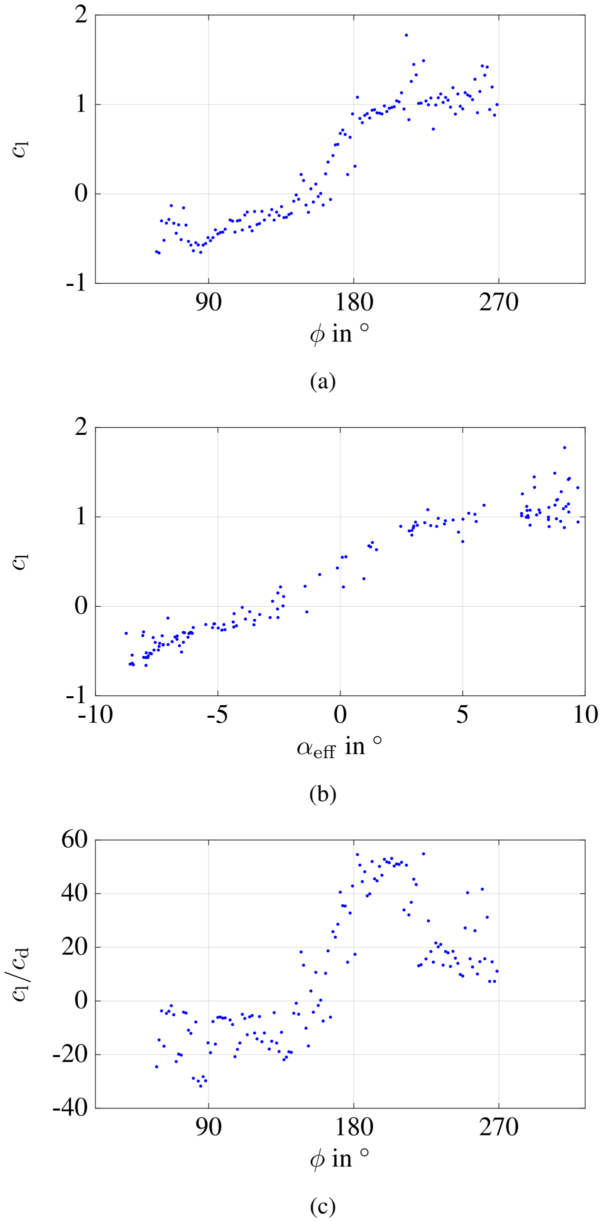


Figure 25: Lift coefficient over phase angle and different effective AOAs as well as lift-to-drag coefficient c_l/c_d over phase angle.

Precision Landing for Fixed-wing UAV using Ultra-Wide-Band Ranging

Gautier Hattenberger*
ENAC, University of Toulouse, France

ABSTRACT

Fixed-wing mini UAVs usually require a large area in order to safely land, eventually directly using the fuselage body as a landing skid. In some cases, the ground surface is not suitable for this type of operation and may damage the aircraft. A common option is to use a net to capture the plane, but the GPS accuracy may not be sufficient to allow a reliable landing in full autonomy. In this paper, we are investigating the use of Ultra-Wide-Band communication modules, used as ranging systems, in order to determine the position of the UAV during its final approach. This information is then used to adjust the trajectory towards the landing net. The focus is made to the calibration procedure, the data fusion Kalman filter to estimate the position of the UAV and the overall performances of the system.

1 Introduction

Fixed-wing mini Unmanned-Aerial-Vehicles (UAV) usually require a large area in order to safely land, eventually directly using the fuselage body as a landing skid. In some cases, the ground surface is not suitable for this type of operation and may damage the aircraft. A common option is to use a net to capture the plane, as seen on Figure 1, but the GPS accuracy may not be sufficient to allow a reliable landing in full autonomy.

Several solutions for precision landing of aircraft are available, mostly relying on vision [1, 2] or Differential-GPS (DGPS). Solutions based on vision usually require an higher computational power and sensors (cameras) that may not be adapted to flights in harsh conditions (such as inside clouds for meteorological studies). DGPS is rather easy to use but can be a bit more expensive. Radar and Lidar are not considered since this type of sensors are most of the time too heavy and expensive to be used on light UAVs. An other popular technology is to use Ultra-Wide-Band (UWB) communication devices that can also provide accurate range measurements [3]. This approach have been already successfully applied on UAVs, eventually in combination with other sensors [4, 5].

The principle of localization based on distance measurements rely on algorithms called trilateration or multilateration depending on the number of measurements [6]. The position can be extracted by direct computation or by using linear algebra like Singular-Value Decomposition or Bayesian filters such as Kalman filters [7, 8].

After describing the parametrization and the reference frame used in this article, the next section will present the trilateration direct method and evaluate its performances for localization of a UAV. Then, based on experimental data, a solution using an Extended Kalman Filter for continuous integration is proposed, eventually taking the turn rate into account to reduce the latency.

Figure 1: Fleet of UAVs for meteorological studies in harsh and remote location. A net is required to land the planes. Courtesy of Greg Roberts from French Meteorological Research Center.

2 Problem statement

In order to build a coordinate frame local to the landing net that would be easy to install for the UAV operators, a first assumption is made on the placement of the anchors used for distance measurements. As it will be shown in section 3.2, a minimum of three anchors are required for this localization problem. If we consider a plane landing at the position (0,0) in a direction facing the x axis, the Figure 2 is showing the situation (top view, xy being the horizontal plane). Each anchor is measuring a distance ($d1$, $d2$ or $d3$) to the plane equipped with a fourth module configured as a tag. The relative positions of the anchors are defined by a lateral separation d_{lat} between anchors 2 and 3 symmetrical along the y axis, and a longitudinal distance to the origin d_{lon} of the third one along the x axis.

The desired landing path for a typical mini UAV of 1 me-

*gautier.hattenberger@enac.fr

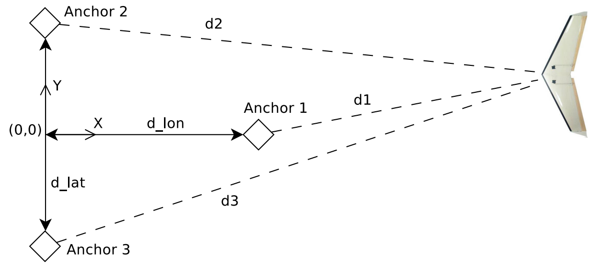


Figure 2: Three anchors and the local coordinate frame.

ter wingspan with a flight speed around 15 m/s is a slope $\gamma_{land} = 10°$, as depicted on Figure 3. This reference path will be used later for the theoretical precision evaluation in section 3.3.

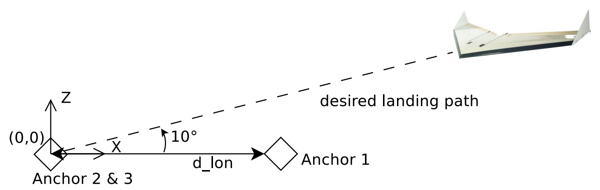


Figure 3: Desired landing path with an angle of 10°.

3 Localization based on ranging

3.1 *Calibration of the ranging devices*

A first step to localization is the calibration and the performance evaluation of the ranging devices. In this study, the *DW1000M* modules from *Decawave*[1] have been used. According to the datasheet, the ranging accuracy is 10 cm, with a maximum distance around 200 meters in direct line of sight. It has been measured up to 250 meters by setting the transmitter and receiver power to the maximum. The calibration process consists in comparing the distances reported by two modules (an anchor and a tag) with ground references placed every 10 meters (up to 100 meter) and match the data with the model $distance = scale * raw - offset$. The result presented Figure 4 clearly shows the excellent correlation of the measured data other the references. In addition, the standard deviation is stable for all distances with an average $\sigma_{avr} = 0.026$, which means that most of the measures (in the range 3σ) are less than 10 cm.

The Table 1 gives the calibration of the three anchors against the tag module. As expected, the scale factors are all close to 1, and the offsets are all less than 1 meter.

3.2 *Direct position from trilateration or multilateration*

The method used to determine the 3D position of the tag based on distances will depend on the number of anchors N,

[1]http://www.decawave.com

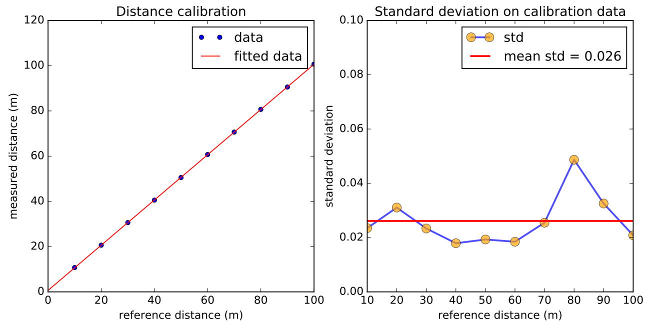


Figure 4: Calibration points and standard deviation of a DW1000 module.

anchor	scale	offset (in meter)
1	0.9999	0.6389
2	0.9972	0.9027
3	0.9979	0.8506

Table 1: Calibration results for the three anchors against the tag module.

assuming that they respect certain constraints.

- If $N = 3$ (and anchors not aligned), a direct computation called trilateration is possible and will lead to an unique 2D position or 2 symmetrical 3D positions. Since in our case the anchors are placed on the ground, the horizontal plane is the symmetry plane and the negative altitude can be discarded.

- If $N \geq 4$ (and anchors not in the same plane), a solution of this multilateration problem can be computed by solving a linear system (as seen in [7]) either by a direct method when $M = 4$ (system is invertible) or with a minimization algorithm when $M > 4$.

Considering operational constraints, the case of multilateration is discarded since it requires, to be efficient, that at least one anchor is not placed in the ground plane. In practice, it means that it shall be placed accurately high enough on a fixed pole. This is not usually possible using the natural landmarks and the poles holding the net in the considered scenario are subject to oscillations, due to the wind in particular.

Thus, the trilateration problem is solved using the following formulas, based on Figure 5, starting with the equations of the spheres:

$$\begin{aligned} d1^2 &= x^2 + y^2 + z^2 \\ d2^2 &= (x-k)^2 + y^2 + z^2 \\ d3^2 &= (x-i)^2 + (y-j)^2 + z^2 \end{aligned} \quad (1)$$

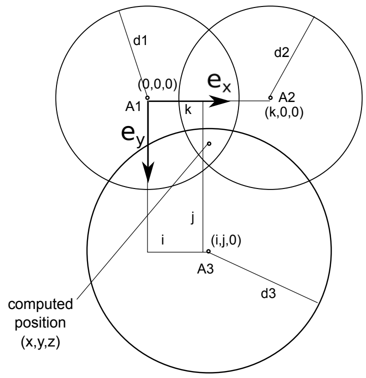


Figure 5: Trilateration problem parametrization.

After rewriting these equations, we have the following position estimate:

$$\begin{aligned} x &= \frac{d1^2 - d2^2 + k^2}{2k} \\ y &= \frac{d1^2 - d3^2 + i^2 + j^2}{2j} - \frac{i}{j}x \\ z &= \pm\sqrt{d1^2 - x^2 - y^2} \end{aligned} \tag{2}$$

The final step is to compute the position P in the original anchors' frame:

$$P = P_{A1} + x\,\widetilde{e}_x + y\,\widetilde{e}_y + z\,\widetilde{e}_z \tag{3}$$

where P_{A1} is the original position of anchor 1, $\widetilde{e}_{[xyz]}$ are the base vectors for the trilateration projected into the original frame, and only the positive z coordinate is kept.

3.3 Precision evaluation

The precision of the position computed with this method is evaluated using monte-carlo simulations. The Figure 6 shows the situation with separations of 5 meters for both d_{lat} and d_{lon}. For each point of the graph, the theoretical measured distances are computed, then a gaussian white noise is added to the measures with the characteristics found during calibration (see section 3.1). Finally the standard deviation σ of the resulting position errors (from 100 simulations) are used to evaluate the precision, assuming that 99% of the points will be in the range of 3σ. The colors of the graph reflect this precision. In addition, the red line correspond to the desired landing path as described section 2.

The first observation is that the precision on the horizontal plane (here at a constant altitude of 20 meters) is about 10 times better than in the vertical plane, where the maximum errors can reach up to 16 meters at 80 meters of the final landing point. The reason for that is that the precision increases when the baseline between the anchors is increasing. On the horizontal plane, there is at least always a baseline of 5 meters in all directions due to the triangular shape of the anchors' locations (see Figure 2). On the vertical plane, because of the

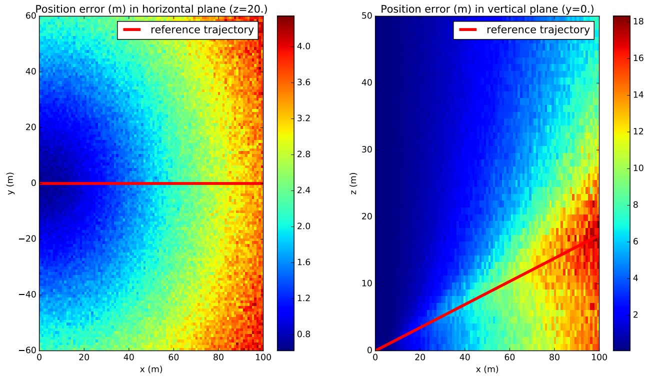


Figure 6: Position errors with d_{lat}=5m and d_{lon}=5m.

followed path, with a small angle of 10°, the actual baseline is much reduced ($d_{lon} * sin(\gamma_{land})$ in this case).

In order to investigate the influence of the lateral and longitudinal separations of the anchors, a graph representing a normalized error score as a function of these two parameters is build. The score is evaluated as the sum of the standard deviation along the reference path:

$$\text{score} = \sum_{\text{along reference path}} \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}$$

The Figure 7 shows the resulting computation with the red lines corresponding to the frontier to have at least a score lower than 40%, 20% or 10% of the worst case.

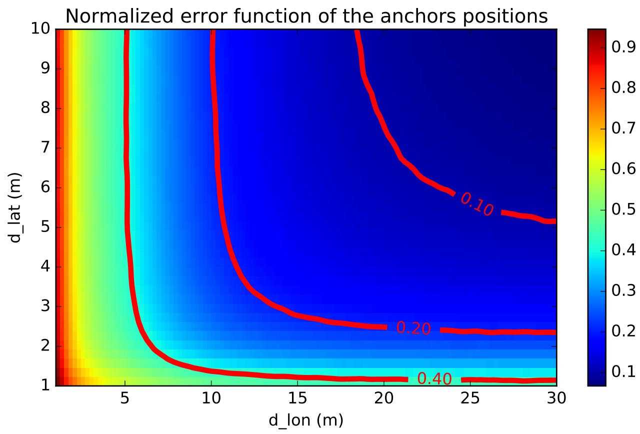


Figure 7: Normalized error score function of d_{lat} and d_{lon}.

The conclusions of this graph are that the lateral distance d_{lat} doesn't influence a lot above 5 meters, while an important value for d_{lon} is required to reach a good overall precision. In fact, a longitudinal distance of 30 meters is needed to have similar results on all axis. This is again rising an operational issue that placing accurately the third anchors at a long distance in front of the others is not easy to achieve and may decrease the final efficiency of the system. The same procedure applied to other anchors geometry have been considered. Placing the anchor 1 behind the net (negative values of d_{lon}) gives similar but slightly lower results. So this solution should be used only if the front position is not possible. Placing the same anchor 5 meters above the ground on top of anchors 2 (e.g. on top of one of the pole holding the net) gives excellent results in terms of precision but shall be discarded for now due to operational constraints as already stated in section 3.2.

For later analyzes, the parameters that will be chosen are 5 meters for both as an acceptable trade-off between precision and operational constraints.

4 Experimental flight analyzes

4.1 *Experimental setup*

Experimental flights have been performed in order to evaluate the performances of the positioning system in dynamic and long range conditions. The used aircraft is a foam based flying wing, equipped with the *Apogee* autopilot from the *Paparazzi UAV*[2] system [9]. The standard GPS receiver have been replaced by a *U-Blox M8P* differential GPS, with its ground base receiver, in order to have an accurate reference trajectory. The nominal airspeed for this plane is around 15 m/s. The Decawave *DW1000M* module used as tag is connected to the autopilot board through an *Arduino* performing the connection to the anchors and computing the distances. The complete setup is shown Figure 8.

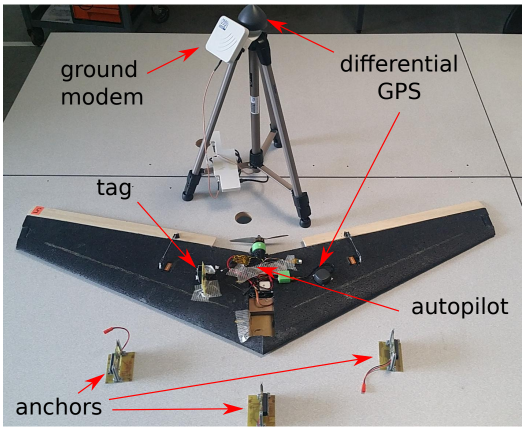


Figure 8: The experimental flying wing equipped with Decawave modules and DGPS system.

4.2 *Direct computation limitations*

The Figure 9 shows the distance reported by anchor 1 while the plane is flying over it about 20 meters above ground. Compared to the distance computed from the DGPS positions after the flight, the correlation is good, but it can be noted that some wrong measurements can occur (at t=1140s) and some over are missing especially far from the anchor. The normal frequency during the flight with 3 anchors was 6.9 Hz. Also, during flights, the maximum range is usually shorter than static tests, with valid measures around 160 meters away, which is still acceptable for the landing procedure.

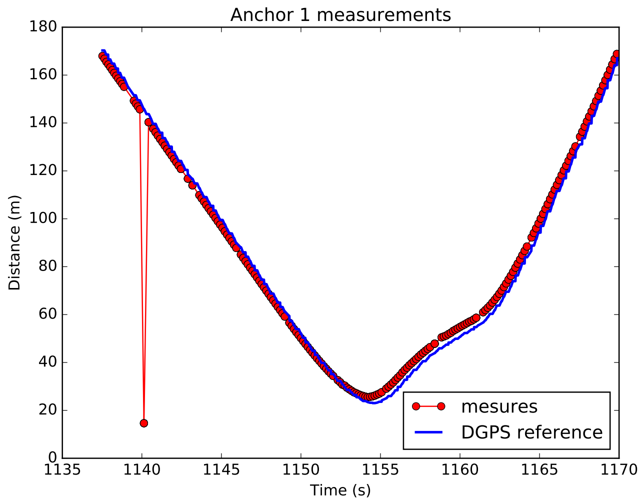


Figure 9: Measurements of anchor 1 compared to DGPS reference.

The main issue of the trilateration direct computation is that it makes the assumption that all three distances are available at the same time. This is not valid during dynamic flight, when the distances are retrieved in sequence while the plane is moving. With an airspeed of 15 m/s and landing in front of a 5 m/s wind speed as during flight test, the distance flown between two measures is about 1.5 meter, which is way above the 10 cm precision used during the static performance evaluation. This issue is illustrated by the Figure 10 showing many wrong position estimations. The reference trajectory coming from the DGPS and projected in the anchors' frame still show a good correlation except for the z axis far from the anchors (at the beginning and the end of the plot).

4.3 *Extended Kalman Filter for continuous correction*

The solution to overcome the problem raised in the previous section is to perform a continuous integration of the data each time a new distance measurement is available. This is typically done with Kalman filters. Since the problem involves non-linear equations that are simple to derive, the most suited filter appears to be the Extended Kalman Filter (EKF), easier to implement than Unscented or Particle filters.

To implement such filter, the model is a second order kinematic model with constant velocity (the acceleration representing the command vector is always null) with position

[2] http://paparazziuav.org

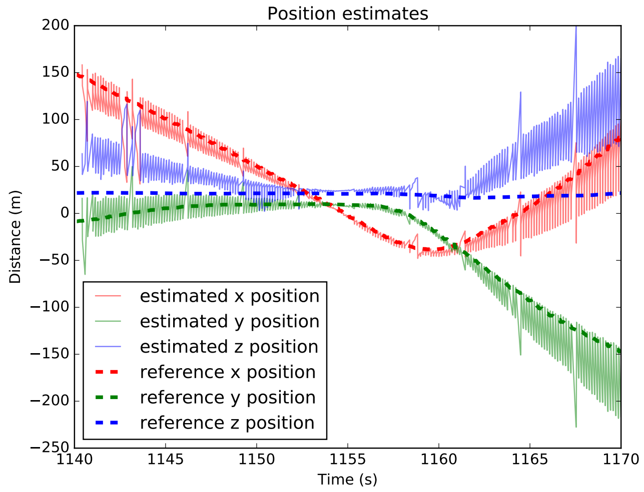


Figure 10: Estimated positions from direct computation.

and velocities as state vector elements:

$$\widehat{X}_k = [x_k \quad y_k \quad z_k \quad vx_k \quad vy_k \quad vz_k]^T \tag{4}$$

In discrete time, the dynamic equation and covariance propagation are thus:

$$\widehat{X}_{k+1|k} = F_k\, \widehat{X}_k = \begin{bmatrix} 1 & 0 & 0 & T_e & 0 & 0 \\ 0 & 1 & 0 & 0 & T_e & 0 \\ 0 & 0 & 1 & 0 & 0 & T_e \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \widehat{X}_k \tag{5}$$

$$P_{k+1|k} = F_k\, P_{k|k}\, F_k^T + Q_k \tag{6}$$

where T_e is the sampling interval and Q_k the process noise.

The observation model h_i corresponds to the distance measurement from each anchor at position $[ax_i\ ay_i\ az_i]$ with $i \in \{1, 2, 3\}$. Then:

$$h_i(\widehat{X}_k) = \sqrt{(x_k - ax_i)^2 + (y_k - ay_i)^2 + (z_k - az_i)^2} \tag{7}$$

and its Jacobian is:

$$H_i(\widehat{X}_k) = \begin{bmatrix} \frac{x_k - ax_i}{\sqrt{(x_k-ax_i)^2+(y_k-ay_i)^2+(z_k-az_i)^2}} \\ \frac{y_k - ay_i}{\sqrt{(x_k-ax_i)^2+(y_k-ay_i)^2+(z_k-az_i)^2}} \\ \frac{z_k - az_i}{\sqrt{(x_k-ax_i)^2+(y_k-ay_i)^2+(z_k-az_i)^2}} \\ 0 \\ 0 \\ 0 \end{bmatrix}^T \tag{8}$$

Finally, at each new measurement z_{i_k} from one of the anchors, we can apply to correction steps of the EKF:

$$K_k = P_{k+1|k} H_{i_k}^T (H_{i_k} P_{k+1|k} H_{i_k}^T + R_k)^{-1} \tag{9}$$

where K_k is the Kalman gain and R_k the measurement noise, and

$$\widehat{X}_{k+1|k+1} = \widehat{X}_{k+1|k} + K_k \left(z_{i_k} - h_i(\widehat{X}_{k+1|k}) \right) \tag{10}$$

$$P_{k+1|k+1} = (I - K_k H_{i_k}) P_{k+1|k} \tag{11}$$

update the state and covariance matrix. Note that this implementation only requires to compute the inverse of a scalar (and not a matrix) to compute the Kalman gain.

In order to have a correct and fast convergence of the EKF, it is necessary to initialize the filter with a state close to its true value. The idea is then to use a direct computation method (either trilateration or multilateration depending on the number of anchors) and then use the EKF to integrate subsequent measurements. The result of this process for the same set of data than Figure 10 is shown Figure 11.

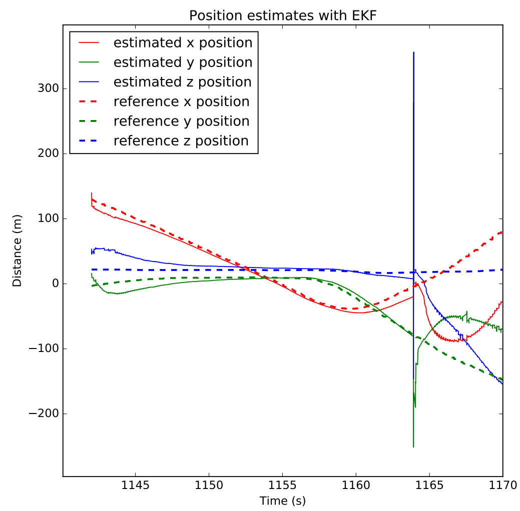


Figure 11: Estimated positions from Extended Kalman filter.

The estimated trajectory is smooth compared to the direct computation and converge to the reference trajectory, especially when the plane is flying towards the anchors (between 1140 and 1160 seconds). After that point, a combination of several effects (high bank angle of the plane while turning, faster speed due to tailwind and especially wrong measurement from an anchor) led to less accurate estimate and even stalling of the filter. It is thus required to implement a proper pre-filter to remove erroneous data, such as peak-removal or median filter.

Note that it is eventually possible to directly integrate the GPS speed norm to the filter as measurement, since the Earth and anchors frame are both fixed relative to the ground, even if their respective axis are different. This might improve the speed estimate of the filter.

4.4 *Improving the dynamic model*

The biggest issue that can be seen figure 11 is the latency in the position estimation when the aircraft is turning. This is coming from the constant speed assumption that is not valid during this phase. The result is that when closing the loop with the trajectory control to stay on the landing axis, this may lead to oscillating or even unstable trajectories. The improvement that can be made is to change the dynamic model to integrate the lateral acceleration as an input. When assuming a coordinated turn, it can be expressed as the product of the speed norm and the turn rate Ω (from attitude estimation filter) around the vertical axis.

$$\left\{ \begin{array}{rcl} \dot{x} & = & V_h\, cos(\psi) \\ \dot{y} & = & V_h\, sin(\psi) \\ \dot{z} & = & V_z \\ \dot{V_h} & = & 0 \\ \dot{\psi} & = & V_h\, \Omega \\ \dot{V_z} & = & 0 \end{array} \right. \tag{12}$$

With this modification and peak filtering, the resulting position estimation finally performs well even during the turn as seen figure 12. Some errors remain when the plane is flying away from the anchors.

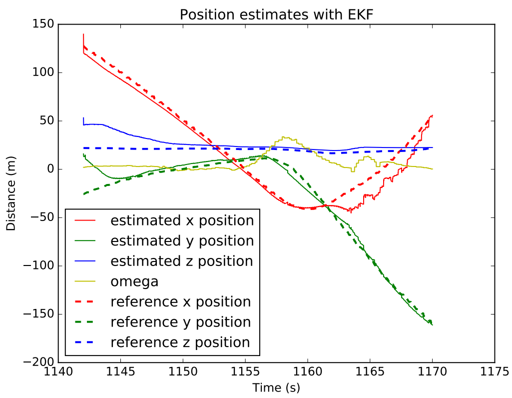


Figure 12: Position estimation with turn rate input.

5 Conclusion

In this article, we have presented an evaluation of the performances of a precision landing system for fixed wing mini-UAVs. The Ultra-Wide-Band technology is offering an easy way to deploy anchors and tags in order to provide accurate distance measurements, and it is compatible with the flight speed of this type of aircraft. Despite that, it is required to apply the correct combination of filters to efficiently estimate the position of the plane in the local anchors' frame. It also have been shown that the optimal location of the anchors are usually conflicting the operational constraints. This approach for local positioning can also be reused in various situations, like indoor flight or precision landing of rotorcraft UAVs.

Acknowledgements

The author would like to thank Enac students Loic David, David Espert, Xin Guo and Adrien Lancelon for their work in the inital study of this problem.

References

[1] V. F. Vidal, L. M. Honrio, M. F. Santos, M. F. Silva, A. S. Cerqueira, and E. J. Oliveira. Uav vision aided positioning system for location and landing. In *2017 18th International Carpathian Control Conference (ICCC)*, pages 228–233, May 2017.

[2] Y. Lin, W. Lu, K. Chen, and J. Guo. Vision-based landing system design for a small uav. In *2015 IEEE International Conference on Consumer Electronics - Taiwan*, pages 496–497, June 2015.

[3] F. Lazzari, A. Buffi, P. Nepa, and S. Lazzari. Numerical investigation of an uwb localization technique for unmanned aerial vehicles in outdoor scenarios. *IEEE Sensors Journal*, 17(9):2896–2903, May 2017.

[4] W. Kong, D. Zhang, and J. Zhang. A ground-based multisensor system for autonomous landing of a fixed wing uav. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1303–1310, Dec 2015.

[5] H. Li, Z. Zhong, W. Kong, and D. Zhang. A fast calibration method for autonomous landing of uav with ground-based multisensory fusion system. In *2015 IEEE International Conference on Information and Automation*, pages 3068–3072, Aug 2015.

[6] J. Sidorenko, N. Scherer-Negenborn, M. Arens, and E. Michaelsen. Multilateration of the local position measurement. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Oct 2016.

[7] J. Yang, H. Lee, and K. Moessner. Multilateration localization based on singular value decomposition for 3d indoor positioning. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Oct 2016.

[8] Garapati kesava harinath and J. Ravi kumar. Implementation and performance evolution of kalman filters for target tracking using bistatic range and range rate measurements. In *2014 International Conference on Electronics and Communication Systems (ICECS)*, pages 1–5, Feb 2014.

[9] Gautier Hattenberger, Murat Bronz, and Michel Gorraz. Using the Paparazzi UAV System for Scientific Research. In *IMAV 2014, International Micro Air Vehicle Conference and Competition 2014*, pages pp 247–252, Delft, Netherlands, August 2014.

Real-Time Landing Zone Detection for UAVs using Single Aerial Images

L. Oyuki Rojas-Perez*[1], Roberto Munguia-Silva[1], and Jose Martinez-Carranza[1,2]

[1]Instituto Nacional de Astrofisica, Optica y Electronica , Luis Enrique Erro 1, Sta. Mara Tonantzintla, CP 72840, Puebla
[2]University of Bristol, Bristol, UK, BS8 1UB

ABSTRACT

In this paper, we propose a methodology for automatic detection of a landing and no landing zone for unmanned aerial vehicles. This framework consists of two processes carried out in a frame-to-frame basis: (1) an aerial depth estimated with a multiple layer Convolutional Neural Network architecture; (2) zone classification carried out with another Neural Network, based on the Inception Convolutional Neural Network, that learns from the aerial depth estimation. The novel aspects in this work are related to the training of these networks. Depth examples associated to aerial images are difficult to obtain since no public datasets of this sort are available. Likewise, diverse examples of landing and no landing zones based on aerial depth estimation are difficult to generate due to flight restrictions in urban areas. Motivated by this, we exploit public datasets meant for autonomous cars and synthetic data generated with simulation. We carried out evaluations using different synthetic datasets to those of training, real images obtained from the Internet, and flights in real scenarios with promising results.

1 INTRODUCTION

Nowadays, deployment of drones in outdoor missions is a common practice. There is an increasing number of applications where drones are piloted or sent to fly autonomously over rural, urban or natural environments, flying several kilometres away from the take-off position or even beyond the line-of-sight. In this context, it might happen that the mission has to be aborted or simply that the drone has to be retrieved due to low battery, which in many modern drones triggers the autonomous function *return home*. However, low battery is exactly a major cause of drone accidents since the battery may run out way before the drone returns. If the drone is kilometres far away, rather than bringing it home, the pilot may decide to take over and try to land it in a safe spot, using the transmitted video to identify an adequate landing zone. Nevertheless, interferance or delay in video transmission could affect such task.

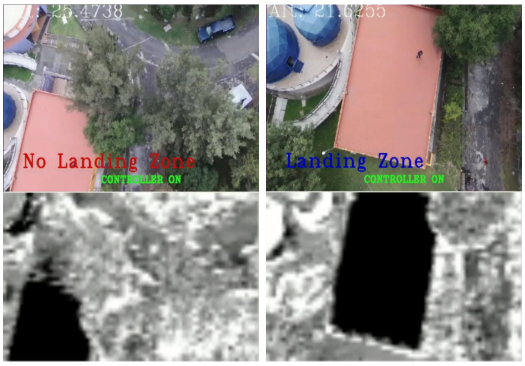


Figure 1: Our proposed system uses two processes, a depth image estimator and a zone classification of single aerial images, aiming at automatically detecting zones that are safe for landing, see video example in: `https://youtu.be/E8TZfErP8F8`

The above calls for a method that enables the drone to autonomously decide whether an observed area is a landing zone or not. Thus, we propose a two-step methodology for landing zone detection where only an onboard monocular camera is used to carry out the detection. The process involves two steps carried out in a frame-to-frame basis. First, a single aerial image captured by the onboard camera s processed by a Convolutional Neural Network (CNN), whose output will be that of a depth image estimation. A second step will grab the estimated depth image and pass it through another CNN network, referred by us as LandNet, that will output one of two classes: *Landing* and *No-Landing*, see Figure 1.

To achieve the above, the first CNN architecture for depth estimation has been trained using the well-known KITTI dataset, where RGB images and corresponding depth data are available. However, the chosen KITTI dataset was recorded with a camera pointing forward and mounted on a car ridding around a city. The scene is far from similar to what is observed in aerial images captured with a camera on board a drone with camera angle pointing downwards. Yet, we argue that the KITTI images and depth data could be exploited

*Department of Computer Science at INAOE. Email addresses: {oyukirojas, carranza}@inaoep.mx

to learn depth associated to texture patterns, in particular for depths in between the 20 and 60 metres, and can be *transferred* to the textures observed from the aerial cameras, thus useful to estimate depth. We followed this approach since there are no datasets available for aerial images with corresponding depth images, something difficult to generate with current stereo or depth cameras that work only for short depth ranges and under artificial light.

The second part of our approach exlploits the idea that depth image estimates belong to a metric space in the 3D world, not in the texture space of RGB images. Hence, we decided to test our depth image estimator on a simulated urban environment where we could obtain depth images. We observed that our depth estimator exhibits an error of up to 10 meters for aerial images captured at 40 or more meters of height. However, the depth estimates appear coherent regarding planar surfaces, which are exactly what we desire to detect as landing zones. Thus, by using simulation, we generate loads of annotated depth images with landing and no landing labels, and train a second CNN aim at generating a model that learns to separate the 3D shape of a landing zone from a no landing zone.

To present our approach, this work has been organized as follows: section 2 presents the related work; section 3 describes our methodology; section 4 presents our results and section 5 discusses our conclusions and future work.

2 Related Work

Autonomous landing problem for UAVs has been widely studied using different approaches and over diverse scenarios. We can find two main scenarios for autonomous landing, those where the selected site is static [1, 2, 3, 4, 5, 6, 7, 8], and others when is in moving (e.g., ships or other kinds of vehicles) [9, 10, 11, 12, 13, 14, 15]. Nevertheless, one element in common in both scenarios is the use of the markers that are placed on the selected landing site.

Among the approaches applied to the problem of autonomous landing, the use of sensors such as Light Detection and Ranging or Laser Imaging Detection and Ranging (LiDAR) [1, 2, 3, 4] provides of point clouds that can be processed aiming at detecing adecuate landing surfaces. For instance, in [3], the authors proposed a deep learning approach, where LiDAR point clouds are classified into a volumetric occupancy by using a 3D CNN architecture. In contrast, other works are opted to use infrared vision information [7, 8, 9, 10]. The works in [9, 10] describe a cooperative system where the objects at the ground form a T-shaped, being their detection used to estimate UAV's pose and performing an autonomous landing. In [7], a system based on an infrared stereo vision is proposed. The system is fixed on the ground and is used to track the UAV's position during the landing process. In [8], an infrared camera array guidance system is used to drive the landing. Although these systems are shown to work, they are tailored for specific zones, needless to say that those using infrared cameras are prone to misbehave in zones with high temperatures. .

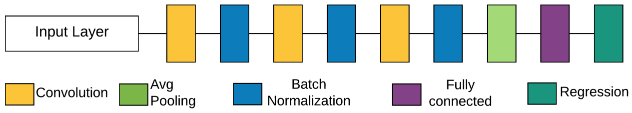


Figure 2: CNN architecture used for aerial depth estimation.

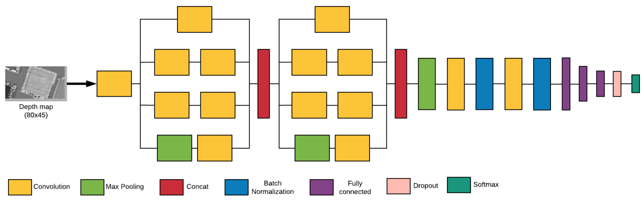


Figure 3: Architecture of the zone classifier based on Inception modules for the feature extraction section of the network.

Other systems for autonomous landing rely only on using RGB imagery transmitted by on board cameras [5, 6, 11, 12, 13, 14, 15, 16]. In [5], GPS in conjunction with vision are used to locate a landing target and land on it, being the vision used to leverage the target detection and recognition.

Finally, another set of works propose to use landmark detection to recognize landing areas. For instance, in [16], the authors propose a CNN architecture based on two efficient nets: YOLO [17] and Squeezenet [18]. The main idea is that of detecting different landmarks, in addition to maintain the efficiency of the detection. In contrast, in this work, we propose to carry out the detection without depending on any visual texture or marks.

3 Methodology

The task of detecting landing zones were divided into two secondary tasks, these tasks are the estimation of aerial depth maps and the classification of possible landing zones. Our approach on solving these tasks is based on machine learning techniques. For the task of aerial depth estimation we designed and trained a CNN following a methodology based on extraction of patches from RGB images. For the problem of zone classification from the depth maps, we propose a classifier based on the Inception modules [19].

3.1 Aerial depth estimation approach

The estimation of depth from aerial images is addressed using a methodology that exploits visual information by processing tiny patches from publicly available datasets imagery. For our purpose we processed images from the KITTI dataset to extract these patches from both RGB images and depth maps. The idea behind this approach is to take advantage of the farthest depth data available in the dataset imagery,

(a) City to generate aerial synthetic images for training.

(b) City to generate synthetic aerial images for evaluation of our approach.

Figure 4: Simulated urban scenarios created in Gazebo 7.

the greatest depth is located at the central part of the image, which is corresponds to the horizon view in the KITTI dataset. The patches used for processing are extracted from the central part of the image and are extracted after dividing the RGB image into equal-size patches. Once the patches are extracted, each patch is mapped to the average depth of the pixel neighbourhood in the depth map, the patches without depth information were excluded. This methodology lead to explore the design of a CNN architecture, which is shown in Figure 2.

This CNN architecture accepts the 25×25 patches as input and is followed by 3 convolutions with a batch normalization, in the final part of the architecture a max pooling operation is performed and a regression layer is used for the estimation of the depth of the corresponding patch. We designed this architecture given that most of the current CNN architectures work with images of a higher resolution to that of our extracted patches.

3.2 LandNet

The aim of the LandNet network is to detect possible landing zones from a given depth map, in our case this depth map is obtained from the mentioned depth estimator. We choose this approach due to the fact that while our depth estimator is good at recovering the shape of scene view it lacks of precision on the estimated depth and it would affect approaches based on geometrical detection of landing zones.

The proposed network accepts as input a depth map with size of 80×45 pixels and is followed by two Inception modules, corresponding to the feature extraction part of the network, see Figure 3. We use these Inception modules to extract different size of features from the depth map and to give the network the capability of learning more details from the depth map. After these two modules, we use a max pooling layer of size $(2, 2)$ followed by two convolutions and batch normalization. In the last part of the network we added three fully connected layers and a final softmax activation function. All the convolutions in our network, including the ones composing the Inception modules, use rectified linear activation. The output of the network is a probability of which of the two classes the actual view corresponds to.

3.3 Datasets

Our datasets consists on images collected from a simulated environment using the Gazebo simulator. We designed two different cities for this purpose, each city is composed by buildings, roads and trees as shown in Figure 4. We used an AR Drone MAV in the simulation for collecting the datasets. The images were collected from the bottom camera of the MAV. These images were feed to our aerial depth estimator to obtain the corresponding depth map for each of the RGB images on our dataset. In order to consider a depth map containing a possible landing zone the estimated map should contain a plain surface, to achieve this we monitored the RGB image and the estimated depth map and when the MAV reached a plain surface the estimated depth was stored into a file, Figure 5(a). The depth maps corresponding to views of trees, mountains or non-plain roofs were considered as no-landing zones, Figure 5(b). A total of 2000 images were obtained from the simulations environment for training and 1000 images for evaluation, with an equal number of samples for each of the two classes.

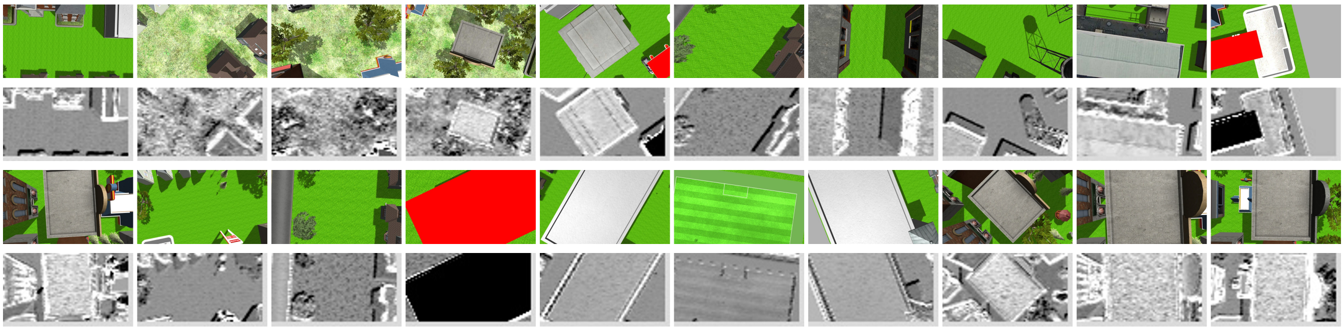
4 Experiments and Results

In this section, we describe the evaluation results, the control architecture employed, the vehicle hardware implemented and the experiments realised in simulation and outdoor environments.

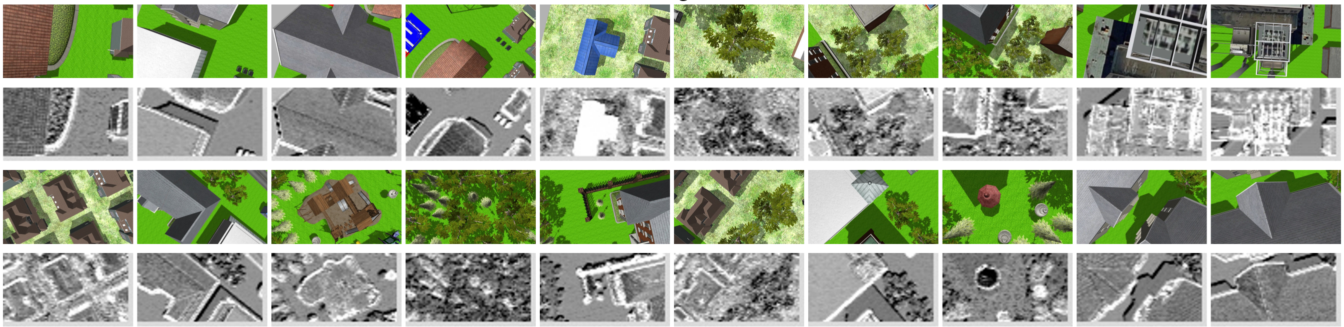
4.1 Evaluation

To evaluate the performance of our landing zone detection, we used two new datasets. Each dataset includes 1000 aerial images, 500 images of landing zones and 500 no landing zones. The first dataset was collected from a second simulated city on Gazebo, see Figure 6(a), and the second dataset was collected from the internet images, Figure 6(b).

We compared the evaluation results of our proposed classifier to the classifier trained with RGB images. For the first dataset, our classifier achieved 82% of accuracy to detect a secure landing zone and 75% to detect no landing zone, Figure 7(a), and the RGB classifier achieved 67% and 72% of accuracy to detect a secure landing zone and no landing zone,

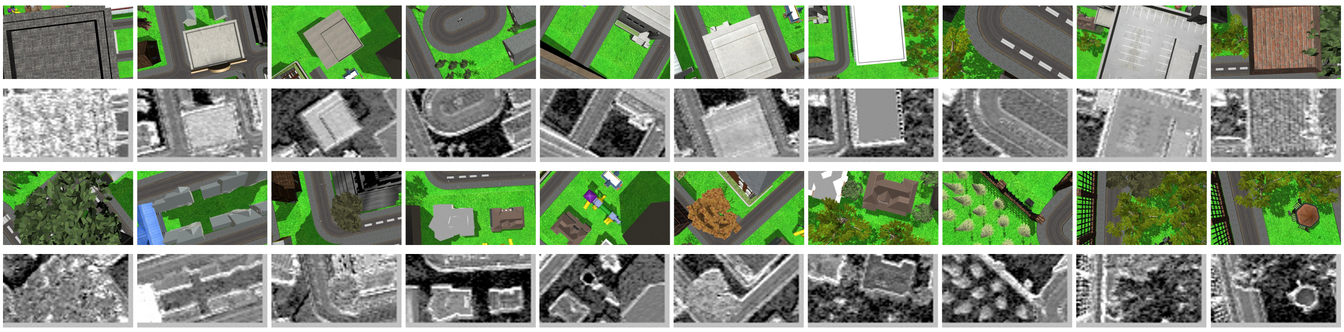


a) Landing Zones.

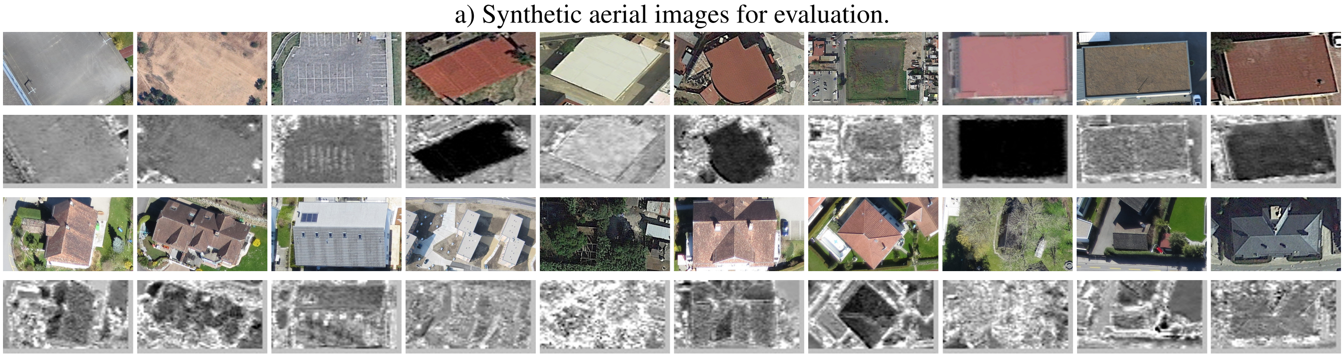


b) No landing zones.

Figure 5: Examples of synthetic aerial images, including their estimated depth image with our approach. Note that planar surfaces, ideal for landing, exhibit similar gray tone. For depth images, the farther the depth, w.r.t. the camera, the darker the gray value.

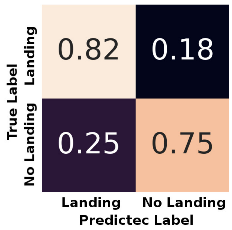


a) Synthetic aerial images for evaluation.

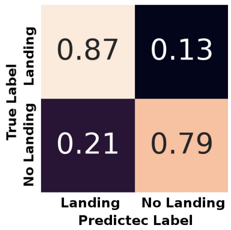


b) Aerial images downloaded from the Internet for evaluation.

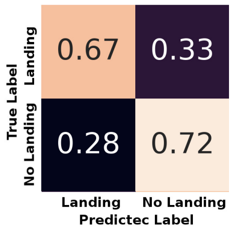
Figure 6: Examples of aerial images, and their corresponding estimated depth, used for evaluation: the first set corresponds to synthetic images generated with the second simulated city, shown in Figure 4.b; the second set corresponds to aerial images downloaded from the Internet.



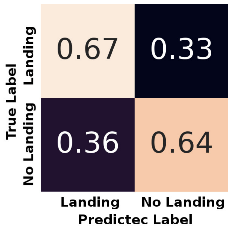
(a) Training: depth; evaluated: synthetic images.



(b) Training: depth; evaluated: Internet images



(c) Training: RGB; evaluated: synthetic images..



(d) Training: RGB; evaluated: Internet images.

Figure 7: Confusion matrix of our CNNs trained with depth images and compared against when trained with RGB images. Better results are obtained with depth.

respectively, Figure 7(c). According to the confusion matrices, our classifier trained with depth maps is more effective to detect secure landing zones compared to the classifier trained with RGB images, this is attributed to the capability of the network for learning depths of planar regions. For the second dataset, our classifier increments by 5% on the accuracy to detect a secure landing zone and by 4% to detect no landing zone, Figure 7(b) and the classifier trained with RGB images decreases its accuracy 8% to detect a no landing zone, Figure 7(d). This confusion matrix proves that our classifier based on depth maps is more stable to identify secure zones to land compared to RGB classifier.

4.2 System Architecture

The proposed landing detection system was implemented in ROS Kinetic Kame and consists of three nodes. One enables the communication with the drone for image transmission from the the drone to the computer and enables the control of the drones from the computer. The second node acquires the RGB image from the drone to obtain the depth image using the described method in section 3.1. Then we used the classifier described in section 3.2, to define if the current zone is a secure zone for landing and publish a flag. In the last node, we implement a controller. This node receives the flag and uses for two options: 1) to continue the navigation until finding a secure zone to land; 2) to keep the drone on hovering and then send a signal to land. To guarantee a secure landing, we use the average of 10 classified frames to decide whether a landing zone is detected or not.

4.3 Experiments in Simulation

We first present experiments in a simulated environment, and we use tum_simulator package and Gazebo 7. The simulated drone navigates upon the city shown in the Figure 4(b). We flew the drone in manual mode at 15 metres in height. The path flight consisted of cover areas to detect possible landing zones when the classifier detects a landing zone the drone stay on hovering and then lands. The Figure 8, shows the simulation test, each figure show the simulated environment, the image from the drone and the estimated depth.

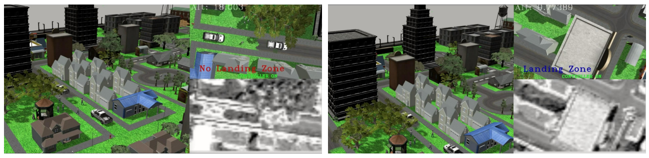


Figure 8: Two illustrative examples of our approach tested in a simulated urban environment.

4.4 Experiments in Outdoor Environment

For these experiments, the tests consisted of flying the drone in a real outdoor environments at 25 meters in height. The drone flew over trees, buildings with flat and spherical roofs. In Figure 9, the images show examples of the detection, whether it is a landing or a no landing zone. When flying over the roof of a building, our system correctly identifies it as a landing zone. Our system exhibit a processing time of $5Hz$, which includes depth estimation and landing/no landing classification.

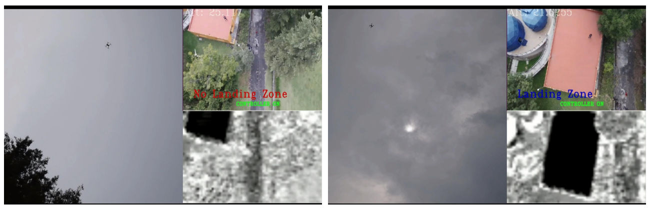


Figure 9: Two illustrative examples of our approach tested in real outdoor, see `https://youtu.be/E8TZfErP8F8`

5 Conclusion

We have presented a method for automatic detection of landing zones for UAVs using aerial images captured with an onboard RGB camera. To address this problem we have used two main ideas: (1) to estimate a depth image from the aerial image; (2) to use the estimated depth images, annotated accordingly, to train a computational model that will learn landing zones and no landing zones. For both tasks, we have implemented two CNNs, where we have used a public dataset from another domain (autonomous vehicles) to obtain training examples for aerial depth estimation, and also using synthetic data generated with a simulator of urban scenes, useful to generate loads of aerial depth images.

We carried out tests with different synthetic data, images from the Internet and during flight missions in real challenging scenarios, obtaining around 80% of accuracy for synthetic and real images. We have also studied the case of when our approach is trained with RGB images instead of depth images, for the landing/no landing detection, obtaining and average of 65% in accuracy, proving the benefit of using depth data rather than texture. In average, our whole system works at $5Hz$, however, we are confident that we can speed up this process by refining our CNN architectures.

As future work, we will improve upon the aerial depth estimation as much as to increase the processing speed.

REFERENCES

[1] M. Whalley, M. Takahashi, P. Tsenkov, and G. Schulein. Field-testing of a helicopter uav obstacle field navigation and landing system. In *Proceedings of the 65th Annual Forum of the AHS, Grapevine, TX.*, 2009.

[2] S. Scherer, L. Chamberlain, and S. Singh. Autonomous landing at unprepared sites by a full-scale helicopter. *Robotics and Autonomous Systems*, 60(12):1545 – 1562, 2012.

[3] D. Maturana and S. Scherer. 3d convolutional neural networks for landing zone detection from lidar. In *2015 ICRA*, pages 3471–3478, May 2015.

[4] Oscar G. Lorenzo, Jorge Martínez, David L. Vilariño, Tomás F. Pena, José C. Cabaleiro, and Francisco F. Rivera. Landing sites detection using lidar data on manycore systems. *The Journal of Supercomputing*, 73(1):557–575, Jan 2017.

[5] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme. Vision-based autonomous landing of an unmanned aerial vehicle. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 3, pages 2799–2804, 2002.

[6] Xiaoming Li. A software scheme for uav's safe landing area discovery. *AASRI Procedia*, 4:230 – 235, 2013. 2013 AASRI Conference on Intelligent Systems and Control.

[7] W. Kong, D. Zhang, X. Wang, Z. Xian, and J. Zhang. Autonomous landing of an uav with a ground-based actuated infrared stereo vision system. In *2013 IEEE/RSJ IROS*, pages 2963–2970, Nov 2013.

[8] Tao Yang, Guangpo Li, Jing Li, Yanning Zhang, Xiaoqiang Zhang, Zhuoyue Zhang, and Zhi Li. A ground-based near infrared camera array system for uav autolanding in gps-denied environment. *Sensors*, 16(9), 2016.

[9] G. Xu, Y. Zhang, S. Ji, Y. Cheng, and Y. Tian. Research on computer vision-based for uav autonomous landing on a ship. *Pattern Recognition Letters*, 30(6):600 – 605, 2009.

[10] G. Xu, X. Qi, Q. Zeng, Y. Tian, R. Guo, and B. Wang. Use of land's cooperative object to estimate uav's pose for autonomous landing. *Chinese Journal of Aeronautics*, 26(6):1498 – 1505, 2013.

[11] Stephen M. Chaves, Ryan W. Wolcott, and Ryan M. Eustice. NEEC research: Toward GPS-denied landing of unmanned aerial vehicles on ships at sea. *Naval Engineers Journal*, 127(1):23–35, 2015.

[12] H. Lee, S. Jung, and D. H. Shim. Vision-based uav landing on the moving vehicle. In *2016 ICUAS*, pages 1–7, June 2016.

[13] Oualid Araar, Nabil Aouf, and Ivan Vitanov. Vision based autonomous landing of multirotor uav on moving platform. *Journal of Intelligent & Robotic Systems*, 85(2):369–384, Feb 2017.

[14] Alexandre Borowczyk, Duc-Tien Nguyen, AndrPhu-Van Nguyen, Dang Quang Nguyen, David Saussi, and Jerome Le Ny. Autonomous landing of a quadcopter on a high-speed ground vehicle. *Journal of Guidance, Control, and Dynamics*, 40(9):2378–2385, 2017.

[15] A. Rodriguez-Ramos, C. Sampedro, H. Bavle, Z. Milosevic, A. Garcia-Vaquero, and P. Campoy. Towards fully autonomous landing on moving platforms for rotary unmanned aerial vehicles. In *2017 ICUAS*, pages 170–178, June 2017.

[16] Leijian Yu, Cai Luo, Xingrui Yu, Xiangyuan Jiang, Erfu Yang, Chunbo Luo, and Peng Ren. Deep learning for vision-based micro aerial vehicle autonomous landing. *International Journal of Micro Air Vehicles*, 10(2):171–185, 2018.

[17] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *The IEEE CVPR*, June 2016.

[18] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, 2016.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE CVPR*, pages 1–9, June 2015.

On the Scaling of Dragonflies

Graham E. Dorrington*
School of Engineering, Royal Melbourne Institute of Technology, Bundoora, Australia

ABSTRACT

Anisopteran dragonfly allometry is discussed. Induced power during hover is found to scale with body mass raised to the ~7/6 power. The possible existence of an upper body mass limit is suggested and a scenario where the maximum load factor for flight manoeuvres decreases with size. Some brief comments are also made about Meganisoptera, bats and birds.

1 INTRODUCTION

Any correlation in the scaling of micro-air vehicle parameters would merely reflect the design tools employed. In contrast, correlations found in the scaling of flying insect (Pterygota) parameters could offer insight into the principles that govern the evolution of their morphology. Dragonflies (Odonata, Anisoptera) are of interest in this regard since their physiology may have evolved over the past ~180 million years. In the present epoch their adult forms have wing lengths, L, varying between ~15 mm and ~85 mm, but it is not known what ecological and/or physiological factors constrain their size. One possibility is that adult upper size limit is constrained by flight power demands.

Suppose that the scaling of any flying creature is exactly isometric. In this idealised case the total wing area, S, scales with L^2 and total body mass, m, with L^3. In stationary hover (outside ground effect), according to simple Rankine-Froude theory, the time-averaged, ideal, induced power is given by $P_{ind} = W^{3/2}/(2\ \rho\ A)^{1/2}$, where W = mg is the total body weight, ρ is the atmospheric density and A is the actuator area which varies with L^2. Consequently, P_{ind} scales with $L^{7/2}$, or with $m^{7/6}$. Since the 'engine' (flight muscle apparatus) may be expected to be less than some maximum feasible fraction of the total body mass and the mass-specific power output must also be constrained, the maximum feasible power output may reasonably be expected to scale with m. At first sight, this divergence in scaling indicates a possible fundamental limit: at some upper size the power required to hover would exceed the engine power available or, alternatively expressed, some upper feasible body mass would be reached. This Flight Power Size Limit (FPSL) hypothesis was similarly proposed by Pennycuick [1] for birds and by Lindhe Norberg & Norberg [2] for bats. Unfortunately, whilst the FPSL is compelling, it is difficult to confirm or disprove.

As introduced above, the FPSL hypothesis is over-simplistic. For dragonflies (and for other flying creatures) a number of complicating factors need to be considered. First, the scaling of parameters is not isometric. Second, the total flight power in hover is also determined by the wing profile and inertial power of the flapping quad-wing configuration. Third, the muscle power output of dragonflies may not be a simple linear function of muscle mass and could be affected by the efficiency of the power production process - which is influenced by heat transfer. Fourth, the maximum power output required by any dragonfly species is unlikely to be determined by the need to perform stationary hover. All species must necessarily perform more power demanding flight manoeuvres during conspecific combat/mating as well as during predator-prey interactions. Male dragonflies also carry females during copula and females carry significant egg loads. In other words, the behavioural repertoire (which strongly influences fitness for natural selection) is dependent on the ability of any species to generate excess power and thrust, i.e. each species must be capable of achieving a maximum thrust NW, where N greater than unity.

All these complicating factors will be partially addressed in the following presentation. Despite many unresolved issues and concerns, it is confirmed that for anisopteran dragonflies P_{ind} does scale with $\sim m^{7/6}$ and the FPSL hypothesis remains plausible and warrants further investigation.

*graham.dorrington@rmit.edu.au

2. ALLOMETRY

In biology, a convenient description of scaling is referred to as "allometry" - the premise that some physiological parameter, y, varies as a simple power of another parameter, x, i.e., $y=x^n$. Although biological systems may have fractal characteristics, there is no proven fundamental rationale for any allometric relation among species in any genus, family, or order. It is therefore important to bear in mind that allometry is just a convenient empirical description of parameter scaling.

Anisopteran dragonfly families include: Aeshnidae, Petaluridae, Gomphidae, Libellulidae and Corduliidae. The allometry of each of these families is different and dependent on the sample chosen [3]. For example, for Aeshnidae, May [3] found that m scales with $L^{2.750}$ and wing area, S, scales with $L^{2.076}$. For a larger sample of Anisoptera from four of the aforementioned families, m scales with $L^{2.586}$ and S with $L^{1.749}$ [3, 4]. It should be noted that across these four families wing aspect ratio increases as size increases, since L^2/S scales approximately with $L^{0.25}$, but within the Aeshnidae aspect ratio remains roughly invariant. This illustrates that caution is needed when comparing samples from different families.

Before determining the scaling of induced flight power, it is worthwhile considering the non-isometric scaling of mass. In order to maintain constant mass-specific induced power, among species of varying size, total body mass would have to scale with L^2, but this is evidently not feasible. Wing mass, m_w, is not a dominant mass, but illustrates the difficulty of such scaling. If m_w scaled with L^2, then large wings would become too flexible (see section 4.4). In a sample of 32 Anisoptera, unpublished data (see acknowledgements) reveals m_w actually scales with $L^{2.826}$. Similar structural arguments apply to the thoracic cage that must withstand muscle contraction forces and to the abdomen that acts as pitch counter-balance to the head and thorax. Some body items, e.g., the compound eyes, might vary in a manner closer to L^2, but the mass of major structural items are expected to vary with $\sim L^3$ in order to maintain a constant structural safety factor. The only possibility to (partially) circumvent this structural scaling demand would be for the load factor, N, to decrease with size.

To date, no evidence has been presented in the literature to show that the dragonfly wing flapping sweep angle, ϕ, varies with scale (in any family) and it will assumed this parameter is invariant hereafter, such that the effective actuator area, A, scales exactly with L^2. Hence for the Aeshnidae, combining the aforementioned allometric relations predicts that ideal induced power P_{ind} scales with $L^{3.125}$, or with $m^{1.136}$. In this case, the scaling constant can be found noting that when L=62.2 mm, m= 1.09 g [4].

3. INDUCED POWER SCALING

3.1 METHOD

In this study, instead of simply combining allometric relations in order to obtain the scaling for P_{ind} (like the previous section), the induced power of individuals of known body mass and forewing length, L, was first calculated assuming the actuator area is directly proportional to L^2. Forewing length was used except in a minority of cases when hind-wings were larger, and then the latter was used to define L.

A sample of 328 individuals with masses ranging from 52 mg and 2710 mg from five families was used: 295 Libellulidae, 26 Aeshnidae, 5 Petaluridae, 1 Gomphidae, 1 Cordullidae. Of these, 314 individuals were measured in the USA (see acknowledgements), 12 were measured by the author in Australia [4] and 2 were measured in New Zealand (see acknowledgements).

3.2 RESULTS

A log-log plot of the calculated P_{ind} values is shown in Fig. 1. It can be seen that P_{ind} scales remarkably closely to with 7/6 power index as expected from isometric relations, i.e. $P_{ind} \propto m^{1.168}$ when the best fit for body mass is $m \propto L^{2.618}$.

Removing all the Libellulidae, reducing the sample to 33 individuals, alters the mass range to 513-2710 mg and the scaling becomes $P_{ind} \propto m^{1.154}$ and $m \propto L^{2.99}$.

Isolating the Libelludidae alone alters the mass range to 52-642 mg and the scaling shifts to: $P_{ind} \propto m^{1.154}$ when $m \propto L^{2.353}$. When 25 *Plathemis lydia* and 12 *Perithemis tenera* are removed from this Libelludidae sample, the scaling shifts to $P_{ind} \propto m^{1.159}$ and $m \propto L^{2.489}$.

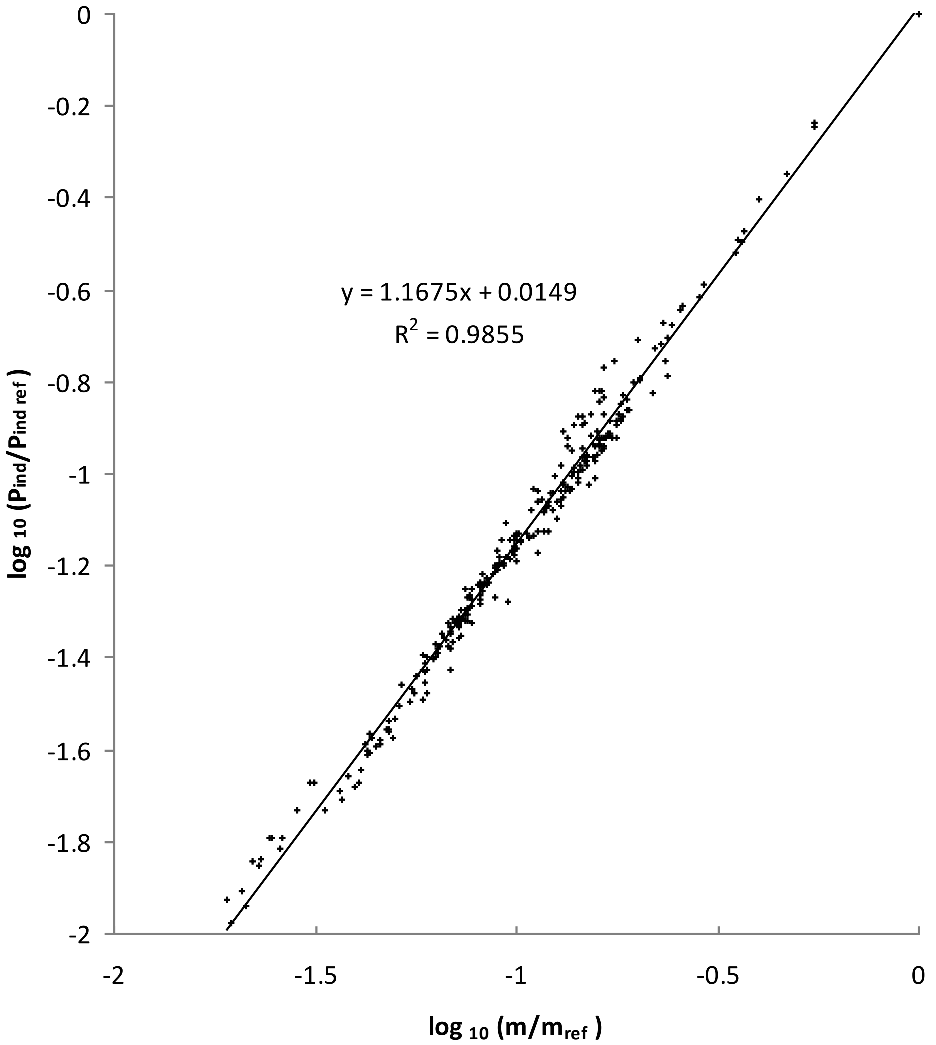


Figure 1 - Ideal induced power of 328 individuals with m_{ref} = 2710 mg

4. DISCUSSION

4. 1 FLIGHT POWER SCALING

The flight power of any animal is the dependent on distinct separable terms that could all scale differently. The total flight power is the sum of: the ideal induced power, P_{ind}; the aerodynamic profile power required to flap the wings $P_{profile}$; the inertial power associated with oscillatory flapping motion, $P_{inertia}$; the sensory and control system power, $P_{sensory}$; finally, the basal metabolic power, P_{basal}, which is the minimum power level during rest. Unfortunately it is not yet possible to arrive at such a confident scaling for all these flight power terms.

Although the kinematics of (insect) flapping has been extensively described in the literature, the scaling of profile power is less often specified. At simplest, for rigid wings, $P_{profile}$ varies in proportion to $\rho S \phi^3 L^3 f^3 C_{Dp}$ where f is the flapping frequency and C_{Dp} is the time-averaged profile drag coefficient which could be a weak function of Reynolds number.

For anisopteran dragonflies May [5] found that flapping frequency scales as $L^{-0.473}$. The unsteady flow over corrugated flapping dragonfly wings is likely to involve leading edge separation, not steady laminar attached flow for which Reynolds scaling would be significant. If Reynolds scaling is ignored, then using May's findings $P_{profile}$ scales $L^{3.33}$ or with $m^{1.297}$. This is similar to the

$\sim m^{7/6}$ scaling of induced power, but the correlation is somewhat poorer and the power index for $P_{profile}$ is strongly influenced by the sample chosen. For a sample with mass proportional to $L^{2.887}$, $P_{profile}$ should scale directly in proportion to P_{ind} at N=1 when May's frequency scaling is used. In order to arrive at a more confident scaling the flapping frequencies and sweep angles of individuals would have to be measured and then used to calculate individual $P_{profile}$ values. It would also be necessary to predict how C_{DP} varies with Reynolds number and other flow conditions in the domain of interest.

The magnitude and variation of the flapping inertia power term, $P_{inertia}$, is also poorly constrained. For a perfect elastic oscillating system the inertial energy invested into each stroke is fully recovered. For a system with zero recovery $P_{inertia}$ will scale with $m_w\phi^2L^2f^3$. If m_w scales with m, then $P_{inertia}$ will vary in proportion to P_{ind} provided that L^2f^3 scales with $m^{1/6}$. Using May's flapping frequency correlation that occurs when m scales with $L^{3.486}$. Given the uncertainty with regard to the recovery factor, it is not unreasonable to assume $P_{inertia}$ also scales in proportion to P_{ind}.

The other power terms listed above will vary with size, but are assumed to be relatively small compared to the sum of P_{ind}, $P_{profile}$ and $P_{inertia}$. Hence the total flight power may simply be reduced to being some constant factor, k_1, greater than P_{ind} at N=1, i.e. $k_1P_{indN=1}$. If the load factor N increases, then the induced power increases to $N^{3/2}P_{indN=1}$ and the total lift force must also increase, i.e., Nm is proportional to $\rho S\phi^2L^2f^2C_L$ where C_L is the time averaged lift coefficient. For fixed m, ϕ and C_L, the flapping frequency must increase as N increases, such that $f = f_{N=1}\ N^{1/2}$, and for fixed C_{DP}, the profile power increases to $N^{3/2}P_{profileN=1}$. Alternatively for fixed m, C_L and f, the sweep angle may increase such that $\phi = \phi_{N=1}\ N^{1/2}$ which also leads to the same result. It is therefore tempting to speculate that the total flight power scales with $N^{3/2}P_{indN=1}$.

4.2. MUSCLE POWER SCALING

Pennycuick & Rezende [6] suggest the maximium direct muscle-mass-specific power output is $P_m/m_m = f(\sigma_m/\rho_m)\Delta L_m/L_m$ where: m_m is the total flight muscle mass; σ_m is the mean dynamic stress in the muscle which is independent of scale; ρ_m is muscle density, also independent of scale; L_m is the muscle length; and ΔL_m is its contraction length. It is reasonable to assume L_m scales approximately with L. To determine the variation ΔL_m of it is necessary to consider the lever mechanism of dragonfly wings. Schilder & Marden [7] show that the main basalar muscle contributing to forewing depression is attached to the wing by an apodeme at a distance, L_{lever}, from the wing hinge point. By measuring this distance in a sample of Aeshnidae and Libelludae, L_{lever} was found to scale with $L^{1.466}$ or with $m^{0.474}$ [4]. Ignoring wing flexing, the basalar muscle contraction distance must be proportional to θL_{lever} where the stroke angle θ associated with basalar muscle is assumed to be invariant with scale (like the sweep angle ϕ). Hence the specific muscle power scales with $fL^{0.466}$. Marden [8] reports that the anisopteran mean muscle mass fraction, m_m/m is ~0.46 and there is some confidence that this fraction is independent of scale. Based on these assumptions the body-mass-specific power output of Anisoptera scales according to, $L^{-0.007}$, i.e. it is found to be approximately invariant with scale [4]. However, it should be noted that this result is at variance with Schilder & Marden [7] who use another commendable approach and conclude that muscle power output varies with $\sim m^{7/6}$. If they are correct, then it indicates that muscle power output could match required flight power requirements with N invariant with scale.

4.3. MAXIMUM LOAD FACTOR SCALING

Dragonflies are not only capable of prolonged near-stationary hover, but also rapid darting manoeuvres including short bursts of high speed horizontal flight or vertical climb, in order to fulfil a variety of behaviour functions. To perform such manoeuvres, the lifting load factor N must be significantly larger than unity. Marden [9] performed artificial load carrying experiments on a variety of insects and found that load factors of N=2-3 are typical. Males and female dragonflies also fly in copula, and there is some evidence that the males (of relatively large species) are capable of lifting females alone without female assistance, which also suggests that load factors of N ~2.2 can be sustained [4].

If the required total flight power is given by $P_{tot} = k_1N^{3/2}P_{indN=1}=k_1N^{3/2}m^{7/6}$, but maximum muscle power output is k_2m, then the maximum feasible size is given by $m_{max} = (k_1/k_2)^6N^9$ when N is assumed to be scale invariant. This relation appears to implausibly sensitive to both the constants $k_{1,2}$ and N. Another possibility is that N is a function of scale: if N scales as $m^{-1/9}$ (i.e. it is just a weak function of size), then the maximum flight power varies linearly with body and muscle mass. In this case there is no body mass limit, but a point is reached when N declines to 1. For example, if existing species have N=2 at m=1 g, then N=1 is reached when m=500 g. Such an extreme limit could only possibly apply to the Palaeozoic Meganisoptera, see section 4.5.

Evidence is clearly needed to substantiate the speculation above. Perhaps the best way forwards is to perform more artificial lift load experiments on a wide mass range of Anisoptera, where wing beat frequency is also recorded. Such experiments would have to be well designed, since some species may have unnatural responses to artificial loading. Use of flight dynamic data might also be revealing, although agility it is not only related to excess thrust-to-weight. Flight agility may improve with reduced wing loading. As size increases, there is a trend for W/S to increase, but the correlation is not strong possibly because some species opt for longer endurance gliding-type flight. As W/S increases, flight speeds in level flight increase assuming the mean cruise lift coefficient is invariant. However, lifting turn radii are likely to decrease as W/S decreases. If larger species are less agile than smaller ones, then it may not be a result of reduced N, but increases in W/S.

At some future date, it may be possible to attach strain gauges to dragonfly wing spars (costa and radius veins) to deduce the forces during flight and thereby find the total flight power output.

4.4. SCALING OF WINGS

Dragonfly wing geometry provides evidence of maximum feasible flight loading. For fixed dynamic pressure, as size increases the aerodynamic forces on the wing would vary with NS and the wing bending moment scales with NL^3. Using simple beam bending theory, in order for the wing bending radius of curvature to vary in proportion to L, for fixed elasticity, the sectional second moment of area of the main wing spars, I_{xx}, would have to vary as NL^4. The main spars are arranged in a corrugated layout at a distance, λ, from the neutral plane, such that $I_{xx} = \Sigma s_i\lambda^2_i$ where s_i is the cross sectional area of each spar (or vein), Fig. 2. By comparing the values of s_i and λ_i for different species at the same relative wing position, it would be possible to find the actual scaling of I_{xx}. The wing mass could also be verified using $m_w = \Sigma s_il_i\tau_i$, where l_i is the effective length of each spar and τ_i allows for spar tapering. Without reliable detailed spar data, however, a simplifying assumption is required to make progress: if the wing profile has some optimum thickness-to-chord ratio, then λ would be proportional to wing chord. For Anisoptera, chord scales with $\sim L^{0.75}$. Assuming the aforementioned result that m_w scales with $L^{2.826}$ and also with the average value of s_iL, it follows that s_i values scale with $\sim L^{1.826}$, i.e. I_{xx} scales with $L^{3.326}$. This indicates that N would have to scale with $\sim L^{-0.6}$ which across a range of 15 to 85 mm is too severe to be plausible. On the other hand, for Aeshnidae chord scales with ~L and I_{xx} scales with $L^{3.826}$ indicating N scales with $\sim L^{0.17}$. This illustrates the necessity to establish an accurate scaling of I_{xx}.

Note also: as size increases, W/S increases, hence dynamic pressure should strictly not be held fixed.

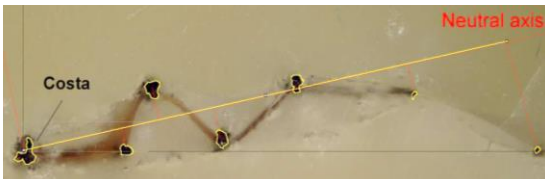


Figure 2 - Cross section through wing of *Petalura ingentissima* set in wax with corrugation depth ~1.1 mm, see acknowledgements

4.5. COMPARISONS

Of course, extant dragonflies are not the heaviest known flying insects. The largest recorded dragonfly body mass is 2.71 g for one female *Petalura ingentissima* [4], whereas, for example, the author measured three individuals of the Empress Cicada, *Pomponia imperatoria,* with masses of 6.84-7.83 g. The past existence of

Meganisoptera with wing spans up to ~710 mm also demonstrates that much larger dragonfly-like insects were feasible in the atmospheric conditions of the Palaeozoic. Although the fossil record of meganisopteran bodies is sparse and estimation of their mass is speculative, one of the largest species may have had a body mass of ~34 g [4], see Table 1.

	Petalura ingentissima	*Pomponia imperator*	*Meganeuropsis permiana*
Mass/g	2.71	7.83	34? [4]
Forewing /mm	83	83	330
Thorax width /mm	12	24	28? [4]
Body length/mm	122	64	?
Forewing area/mm^2	1273	2282	~11,000

Table 1 – Comparison of large insects

Disregarding atmospheric changes, it seems unlikely that any extant dragonflies are close to any FPSL. Nevertheless, it is still possible that selective pressures presently limit extant dragonfly size. If predation of dragonflies by birds is a strong selective pressure, then the ability to evade predation during flight becomes important, i.e. decreased N would reduce fitness. Without this pressure, during the Palaeozoic (before the emergence of birds and pterosaurs), large meganisopterans may have been relatively sedate fliers with limited load lifting capability [4]. If this speculative viewpoint is correct, then collaborative evidence might be sought in wing geometry parameters: compared to extant dragonflies, meganisopteran wings would have had relatively thinner spars and/or the relative wing corrugation depth would be reduced.

Some comparisons with the possible FPSLs for extant birds and bats are also worth mentioning. Bird species that are capable of low specific power soaring are much larger than species that depend on prolonged hovering. Altshuler *et al*. [10] studied the load lifting capability of 677 individual hummingbirds representing 75 taxa with masses ranging from ~2.2 to ~12 g, at four different altitude settings. They found that wing beat frequency varies with muscle mass according to m^{-a} where the index a = 0.43-0.591, and total lifted mass varies according to m^{b}, where b =0.8558-0.6474. As they point-out, the latter result is in contrast to Marden's proposal for universal isometric relation between total lifting load and muscle mass [11].

Lindhe Norberg & Norberg [2] propose a FPSL limit of ~1.6 kg for bats. For bat body masses between 35 and 700 g, they found flight power varies as $m^{1.18}$ (close to the isometric expectation), but wing beat frequency varies with $m^{-0.27}$ and muscle power output varies with $m^{0.73}$. This decline in specific muscle power with size may result from energy limits influenced by heat transfer, i.e. maximum permissible body temperature.

5. CONCLUSIONS AND RECOMMENDATIONS

The scaling of dragonfly ideal induced power shown Fig. 1 agrees remarkably well with the $m^{7/6}$ allometric relation predicted for birds [1] and bats [2]. If profile power and inertia power also scale in this manner (and preliminary findings indicate this is possible), whilst muscle power output scales linearly [4, 6], then it's possible that the upper mass of dragonflies is constrained. However, such a limit must be established at the maximum loaded flight power condition, and it may be a 'soft' selective limit that is dependent on the possible decline of the load factor N with size. It is therefore recommended that artificial lift load experiments, like those performed by Marden [9] on insects and Altshuler *et al* [10] on hummingbirds, need to be performed on a wide mass range of Anisoptera species to determine the scaling of N with size. In such experiments wing beat frequency needs to be recorded to arrive at an allometry for total flight power at the maximum loaded condition. Another promising method to determine the scaling of N is to examine the variation of wing structural geometry parameters, and/or by possibly strain gauging wing spars to obtain in-flight load measurements. With advances in micro-air-vehicle technology, in-flight data logging should be feasible.

Wood [12] states "biology is a useful tool" for the development of flapping wing micro-air vehicles (as small as 60 mg), but equally such engineering efforts could be useful in our attempts to better understand the evolution and physiology of flying insects.

ACKNOWLEDGEMENTS

Much of the data used in section 3 was kindly supplied by Wade Worthen and Ruud Schilder (whose data also included wing masses). Two Uropetala (Petaluridae) specimens were kindly measured by Mark Galatowitsch in New Zealand. Geoff Monteith measured the mass of the heaviest Petalura (section 4.5). The image shown in Fig. 2 was obtained by Suk Yeo and Pengtu Zhang during 4th year undergraduate projects with the support from the RMIT Microscopy and Microanalysis Facility.

REFERENCES

[1] Pennycuick, C. J., Power requirements for the horizontal flight in the pigeon, *Columbia livia*, Journal of Experimental Biology, 49, 527-555, 1968.

[2] Lindhe Norberg U. M. & Norberg, R. Ä., Scaling of wingbeat frequency with body ,ass in bats and limits to maximum bat size, Journal of Experimental Biology, 215, 711-722, 2012.

[3] May, M. L., Allometric analysis of the body and wing dimensions of male Anisoptera, Odonatologica, 10 (4), 279-291, 1981.

[4] Dorrington, G. E., Heavily loaded flight and limits to the maximum size of dragonflies (Anisoptera) and griffenflies (Meganisoptera), Lethaia, 49, 261-274, 2016.

[5] May, M. L., Wingstroke frequency of dragonflies (Odonata: Anisoptera) in relation to temperature and body size, Journal of Comparative Physiology, 144, 229-240, 1981.

[6] Pennycuick, C. J. & Rezende, M. A., The specific power output of aerobic muscle, related to the power density of mitochrondria, Journal of Experimental Biology, 108, 377-392.

[7] Schilder, R. J. & Marden J. H., A hierarchical analysis of the scaling of force and power production by dragonfly flight motors, Journal Experimental Biology, 207, 767-776, 2004.

[8] Marden, J. H., Bodybuilding dragonflies: costs and benefits of maximising flight muscle, Physiological Zoology, 62 (2), 505-521, 1989.

[9] Marden, J. H., Maximum lift production during takeoff in flying animals, Journal Experimental Biology 130, 235-258, 1987.

[10] Altshuler, D. L., Dudley, R., Heredia, S. M., & McGuire, J. A., Allometry of hummingbird lifting performance, Journal of Experimental Biology, 213, 725-734, 2010.

[11] Marden, J. H., Scaling of maximum net force output by motors used for locomotion, Journal of Experimental Biology, 208, 1655-1664, 2005.

[12] Wood, R. J., The first takeoff of a biologically inspired at-scale robotic insect, IEEE Transactions on Robotics, 34 (2), 341-347, 2008.

Autonomous Bird Deterrent System for Vineyards using Multiple Bio-inspired Unmanned Aerial Vehicle

Z. Wang* and KC Wong
University of Sydney, Australia

ABSTRACT

A novel bird deterrent system using multiple Unmanned Aerial Vehicles (UAVs) for vineyards is being proposed. Bird damage in agriculture is a significant and long-standing problem globally. A successful bird deterring system must be effective and autonomous to eliminate cost associated with human operator. In this paper, we derive the hardware requirement for such a system from experimental data, as well as a bird deterring strategy to enable autonomous operation. The hardware and strategy are first tested under manual control to assess effectiveness. The problem of trajectory planning for UAVs is formulated as a model predictive control problem. Models of the bird detection sensor, the bird behaviour, the UAV dynamics and the environment are estimated using experimental data. Occupancy grid map is used to represent the state of the environment, and this map is used to plan the optimal bird deterring trajectory for UAVs. Preliminary results from the simulation indicated that a 40-hectare vineyard can be protected by two UAVs.

1 INTRODUCTION

Managing pest bird damage in agriculture is a challenging problem because of the scale of agriculture sites and unpredictability of wildlife. In Australia, around AU$300 million worth of commercial crops are lost due to pest bird damage [1]; the estimates in the United States may well exceed US$4.7 billion [2]. Many methods have been developed, yet there are only a few effective but expensive methods [3]. Wine grape is one of the most vulnerable commercial crops to bird damage. Netting is the most common methods deployed in vineyards. However, the cost of netting increases as the size of the vineyard increases, making this method too expensive for large vineyards.

With the fast development of UAV and autonomous technologies, there are increasingly more interests in using UAVs for bird damage control among researchers and grape growers. UAVs have the advantage of traversing a large agriculture property in a relatively short period of time compared to ground vehicles. They are also not constrained by rough terrains commonly found in agriculture properties.

*Email address: zwan7346@uni.sydney.edu.au

Figure 1: A photo of the prototype UAV.

The aim of this research is to develop a autonomous UAV bird deterring system for agriculture. The problem in vineyards will be investigated first. There are many challenges to be addressed, including finding the most effective scaring elements; determining the appropriate sensors for bird detection; the ground-to-air communication; and developing the deterring strategy. Such a system will also be desirable in other situations where bird populations may cause damage or are nuisance. These examples include but not limited to: airport, chemical spill sites, aircraft hangars, trains stations and private spaces.

2 RELATED WORK

Natural predatory birds are most efficient at deterring pest birds. A trial conducted on a New Zealand vineyard saw the grape damage reduce by 95% after introducing the New Zealand Falcon in the region [4]. However, hoping for an eagle to appear every time the pest birds are coming is unrealistic. Many commercial solutions and published researches that utilise UAVs to mimic predators exploit neophobia (fear of novel objects) in the pest birds [5]. These UAV methods are indifferent to conventional scaring methods (e.g. scare crows, loud and sudden noises from speakers) as they may also suffer from habituation. Habituation is where the pest birds learn that the UAVs are not real threat and stop associating the UAVs with danger. To avoid habituation, we need to understand the triggers for a long term anti-predatory behaviour. The most important lesson learned from literature is that birds obtain information about predation risk from each other [6]. Birds typically produce anti-predator vocalisations (also known as alarm and distress calls) when a natural predator is spotted. Birds can learn about new predators if a real threat is accompanying the calls [7]. This is the approach implemented in the prototype system.

An autonomous system must also know where the birds are. Bird detection has always been a challenging problem. The birds are not only naturally camouflaged, they are always morphing during flight as well. The problem can be however simplified in this application as the exact number of birds is not important for the decision process, only pest birds that move in large flock are of concern to the vineyard growers.

An appropriate trajectory planning algorithm is also required for autonomous operation. There are many uncertain variables in the environment, such as the bird location and the bird behaviour. Occupancy grid mapping is one of the most appropriate algorithm for this system, it is commonly implemented in 2D autonomous vehicle search and pursuit problems, where the targets are not stationary and sensor data are not entirely reliable [8]. The bird deterring problem can also be formulated as a 2D problem because all the grapes to be protected are located at ground level. The UAVs can be operated at a unique fixed altitude near ground level to avoid collision.

3 System Overview

The following system hardware requirement is proposed based on extensive flight trials. Experiments were conducted in multiple vineyards located in south-east Australia [9]. The autonomous bird deterrent system being proposed consists of four sub-systems. They are the bird detection system; perceived predation risk generator system; flight control system; and ground control system. The organisation and communication between these sub-systems are illustrated in Figure 2.

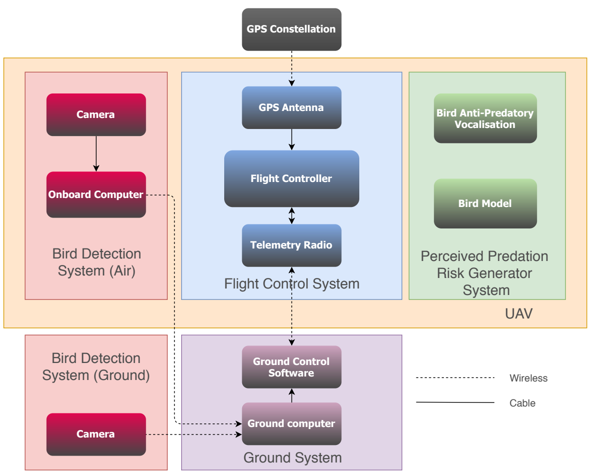


Figure 2: System diagram of the proposed UAV bird deterrent system.

3.1 Bird Detection System

The bird detection system consists of cameras on both the UAV platform and the ground. The camera on the UAV platform enables active tracking of the bird flock, whereas the ground camera can provide early warning of a bird attack. It is not necessary and inefficient energy-wise to have a UAV deployed all the time. The ground camera can be therefore used to decide whether a UAV should be deployed. The ground camera feeds the captured frames directly to the ground system computer for image processing. The frames captured by the air camera on the other hand are fed to an on-board computer for processing. The detection results are then transmitted to the ground computer using wireless communication.

3.2 Perceived Predation Risk Generator System

As mentioned in Section 2, the UAV should produce anti-predator vocalisation, as well as the source of the threat. We have chosen the combination of bird distress calls and a bird model to achieve this goal.

3.3 Flight Control System

The flight control system provides basic stabilisation, and more importantly, position control capabilities. The local position must be known for autonomous operations. GPS is one of the simplest methods, since the only additional hardware required is a GPS antenna. However, GPS lacks the accuracy other vision based methods have. High accuracy position control is always desirable for navigation in cluttered environment. Vineyards on the other hand are usually very open. Therefore, position control relying on GPS is sufficient for the problem.

3.4 Ground Control System

The ground control system is essentially a computer that processes videos from ground cameras for bird detection; runs path planning algorithm; communicates decisions with the flight control system; and monitors UAV status such as altitude, position and battery level through a ground control software.

3.5 UAV Platform

The appropriate UAV platform is determined to be a high endurance multirotor. Multirotors have many advantages over other UAV platforms. The ability to hover and to take-off/land vertically greatly reduce the burden on ground infrastructure. Multirotors are also very simple mechanically, which reduce the risk of failure. The only short-coming is the much lower endurance compared to fixed-wing UAV. However, a typical bird deterrent mission is less than 10min as we discovered in our trials.

4 Deterring Strategy

A flow chart of the decision process is shown in Figure 3. When the system finishes initialising, the ground bird detection system is activated to detect pest birds. In the event of a detection, the UAVs are launched. A trajectory is immediately planned by the ground control system for the UAVs to follow based on the current knowledge of bird location.

As the UAVs execute the mission, the ground control system constantly monitors the battery level of the UAVs. The UAVs are commanded to return home and land immediately if the battery level is lower than the safety threshold. Redundant UAVs are initialised if all other UAVs are insufficiently charged and birds have not left the vineyard. The on-board bird detection system simultaneously updates the bird location for the ground control system. All UAVs are sent home for landing once the birds are sufficiently far away from the vineyard.

5 Manual Flight Experiment Results

A series of manual flight trials were performed at multiple vineyards in south-east Australia to assess the effectiveness of the proposed system [9].

5.1 Experiment Set-up

In the manual flight experiment, all operations and decision making in Figure 3 were carried out manually by UAV operators. A multirotor UAV, as shown in Figure 1, was manually flown to deter pest birds. Birds were detected, and their response were recorded by observers with binoculars on the ground; perceived predation risk generator system was turned on manually at UAV launch; UAV position was controlled from the ground using a remote control transmitter; the relative distance between the bird and the UAV was estimated using the GPS data from the flight controller. The UAV position control was achieved by the flight controller Pixhawk running PX4, and the dedicated GPS module mRo U-Blox M8N GPS [10].

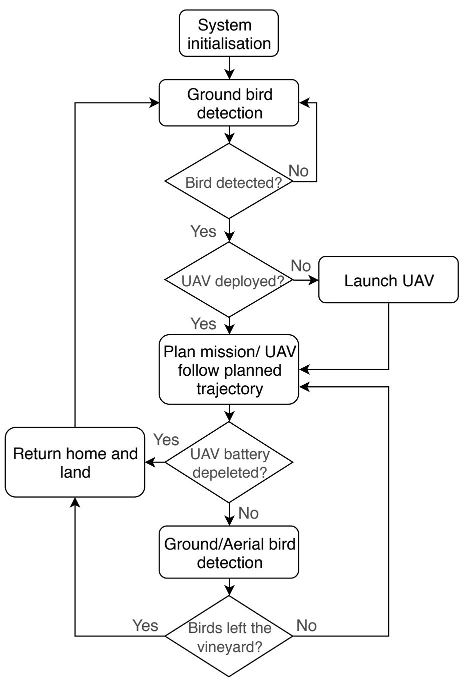


Figure 3: The decision process of the system.

A piezoelectric tweeter was used to broadcast bird distress call. A piezoelectric tweeter was a better choice due to the louder volume at higher frequency compared to a magnet-driven speaker with similar size and weight. The source of the predation risk was a bird model mounted upside-down underneath the multirotor UAV, as shown in Figure 1.

5.2 Bird Response to UAV

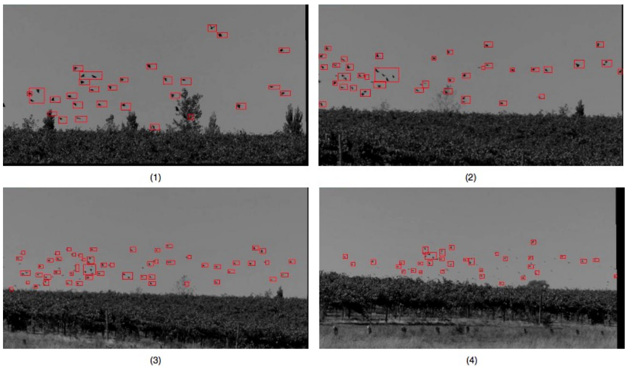
On average, a 10min flight was sufficient to deter all pest birds off a 8 hectare vineyard. Some birds started fleeing 450m away from the UAV, all birds fled the initial location when the UAV was 50m away from them. The targeted bird flocks did not return at least 2 hours after UAV flight. We also determined the birds were only interested in feeding during early morning (6:00-10:00 AM) and late afternoon (4:00-7:00 PM) regardless of the presence of UAV.

The implication of these results was that it was not necessary for the UAV to chase after the birds directly. Instead, the UAV could be treated as a source of influence with a finite radius of effect. As a result, the UAV did not need to be a high speed and high manoeuvrability platform. Furthermore, the UAV did not need to operate throughout the entire day. The birds were only active 7 hours a day, and they did not return for at least 2 hours after a 10min UAV mission. This indicated that a high endurance UAV was not necessary, as there was plenty of time between the flights for battery recharging.

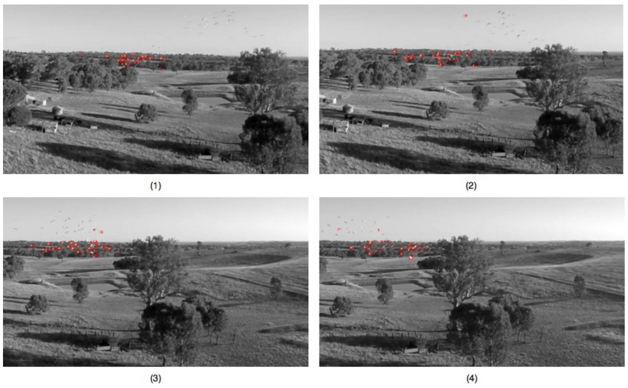
5.3 Bird Detection Results

While the UAV was operated manually, we took the opportunity to test the bird detection algorithm during the trials. The proposed algorithm utilised FAST (Features from Accelerated Segment Test) algorithm [11] to compensate global motion between consecutive frames. The pixel change between the two frames was then analysed with background subtraction using Gaussian mixture models [12] to isolate the actual birds from noises, such as moving leaves.

For bird detection in the air, the computer vision algorithm was implemented on the Raspberry Pi 3 Model B (1.2GHz CPU and 1GB of RAM, running Ubuntu MATE 16.04) [13] and the Raspberry Camera Module V2 [14]. For bird detection on the ground, a Panasonic DMC-FZ200 camera was used. The videos were processed on a Laptop running macOS 10.13.6, with a quad core 2.7GHz Intel Core i7 CPU and 16 GB of RAM. Both systems were able to process the incoming 720P 30FPS footage in real time. Figure 4 shows example frame from the processed footages. The algorithm detected all birds in the frame if the contrast between the birds and the background was high, as shown in Section 5.3; only 60% of the birds were detected in Section 5.3 since the contrast between the sky and the white cockatoos was very low. But the results were sufficient to determine the direction of the flock centroid.



(a) Detection result on ravens from ground camera



(b) Detection result on cockatoos from UAV camera

Figure 4: The proposed bird detection algorithm implemented. Detected birds are bounded by red boxes.

6 Towards Autonomous System

The manual flight experiment demonstrated the effectiveness of the UAV at deterring pest birds and the viability of the deterring strategy. A path planning algorithm is proposed in this section.

6.1 *Environment Model*

As discussed in Section 2, occupancy grid maps are useful when the system is not entirely confident about the target location, and the information is only relevant in 2D. The vineyard area to be protected can be represented by an 2D area that consists of cells of uniform size in the x dimension between x_{min} and x_{max}; and in the y dimension between y_{min} and y_{max}. The spatial domain M of the occupancy map can thus be defined by Equation (1).

$$M = \left\{ \bar{c} \;\middle|\; \begin{array}{l} \bar{c}_x \in [x_{min}, x_{max}] \\ \bar{c}_y \in [y_{min}, y_{max}] \end{array} \right\} \quad (1)$$

Each cell in the occupancy map is located by its coordinates $\bar{c}$. The occupancy map is then defined by a scalar number $k \in [0, 1]$ to each cell $\bar{c} \in M \subset \Re^2$ at a certain time step $t \in \Re$. The scalar number k is the probability indicator for the bird existence at each cell, $k = 1$ represents the system is 100% confident birds are located in that cell, and vice versa

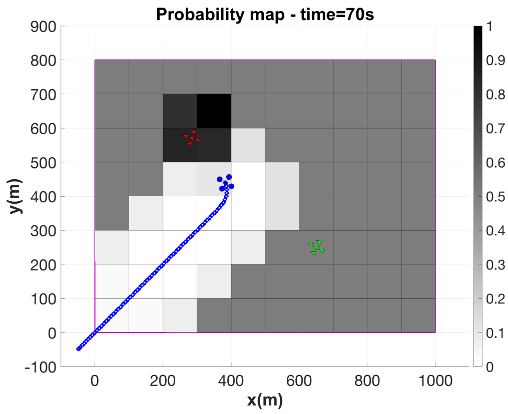


Figure 5: Probability map example.

for $k = 0$. The probability is updated at each time step by information from the bird detection system. This occupancy map is therefore a probability map of the target, an example of the map is shown in Figure 5. In the map, each bird flock is represented by 5 small triangles arranged in a cross (red and green in Figure 5). The UAV is represented by a small triangle surrounded by 4 circles (blue in Figure 5). The path taken by the UAV is indicated by a trail of markers corresponding to the UAV colour. A trail of low probability cells are visible along the UAV trajectory. The 4 cells adjacent to the red target have higher probability as it enters the UAV's sensor field-of-view.

Furthermore, the probability is time varying, it approaches a non-zero nominal value if no sensor information is available. It reflects the fact that the system's confidence about a cell gradually decreases. It also accounts for the possibility of birds returning to previously treated area. The probability approaches the nominal value k_{nom} according to Equation (2).

$$k(t+1, \bar{c}) = \tau_{prob} k(t, \bar{c}) + (1 - \tau_{prob}) k_{nom} \quad (2)$$

$\tau_{prob} \in [0, 1]$ is a time constant that dictates the rate at which k approaches k_{nom}.

6.2 *Sensor Model*

To model the camera, a circular sector of radius r_{sen} and angle θ_{sen} is placed at the centre of the simulated UAV, such that if no bird flock is inside the circular sector, all cells covered by the cone are assigned k_{low}. If any bird flock is inside the circular sector, cells within r_{unc} of the bird flock are assigned k_{high} as an estimation of the sensor uncertainty. This is illustrated in Figure 6. Ground sensor model can be estimated in a similar fashion.

6.3 *UAV Model and Optimal Trajectory*

The UAVs in the simulation have simple second-order dynamics based on the performance of the multirotor used in the manual experiments. The optimal trajectories for the UAVs are the trajectories that can minimise the probability of the entire map while also satisfying the constraints of the UAVs.

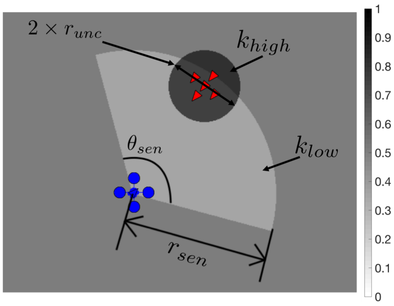


Figure 6: Sensor model illustration.

The algorithm first searches for the optimal cell the UAV is able to reach in the next n time steps. The set of cells C reachable by the UAV in n time steps are first selected according to the UAV's maximum velocity and turning rate. The probability states of these cells $k(t+n, \bar{c})$ are then predicted according to Equation (2). Multiple UAVs can achieve co-operation by taking other UAV's movement into $k(t+n, \bar{c})$ prediction. The problem then becomes a optimisation problem, described in Equation (3).

$$\underset{\bar{c} \in C}{minimise} \quad k(t+n, \bar{c}) + \alpha \cdot f_d(\bar{c}) + \beta \cdot f_h(\bar{c}) \qquad (3)$$

The functions f_d and f_h compute the distance and the heading change required to reach the cell at $\bar{c}$. α and β are weights used to penalise distance and heading change to ensure smooth and efficient trajectories. This optimal cell is the next way-point for the UAV. The current strategy to find the optimal cell is by brute force. This will be improved in the future by employing a proper optimisation algorithm.

6.4 Bird Behaviour Model

The state of the bird x_{bird} is simplified to only position, velocity and heading, and it follows a simple second-order dynamics estimated from observed bird behaviour. We assign a second scalar number $i \in [0,1]$ to each cell $\bar{c} \in M \subset \Re^2$ at a certain time step $t \in \Re$ to represent the birds' interest in visiting each cell. An example of the interest map is shown in Figure 7. The interest value gradually decreases to 0 at the UAV centre as indicated by the lighter cells in the figure.

To account for the likelihood of birds returning to treated area, i approaches a nominal value i_{nom} according to Equation (4) in a similar fashion as k. $\tau_{interest}$ is the time constant that determines the speed at which i approaches i_{nom}.

$$i(t+1, \bar{c}) = \tau_{interest} i(t, \bar{c}) + (1 - \tau_{interest}) i_{nom} \qquad (4)$$

The simulated birds move in the occupancy grid such that they maximise the interest in their surrounding cells. The optimisation problem is now defined by Equation (5).

$$\underset{\bar{c} \in C}{maximise} \quad i(t+n, \bar{c}) + \alpha \cdot f_d(\bar{c}) + \beta \cdot f_h(\bar{c}) \qquad (5)$$

Similarly, the optimal cell is currently searched by brute force. An appropriate optimisation algorithm such as evolutionary algorithm will be investigated in the future.

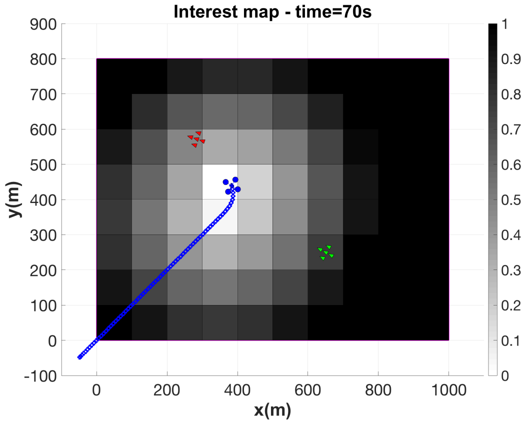


Figure 7: Interest map example.

7 Simulation Results

Important simulation parameters are summarised in Table 1. Two flocks of birds (red and green) and two UAVs (blue and yellow) were initialised. No ground sensors were simulated, knowledge of the bird existence was assumed. According to observed bird behaviour, i is set to 0 within 50m of any UAVs, and i increases linearly to 1 at 450m away from any UAVs as a conservative estimate.

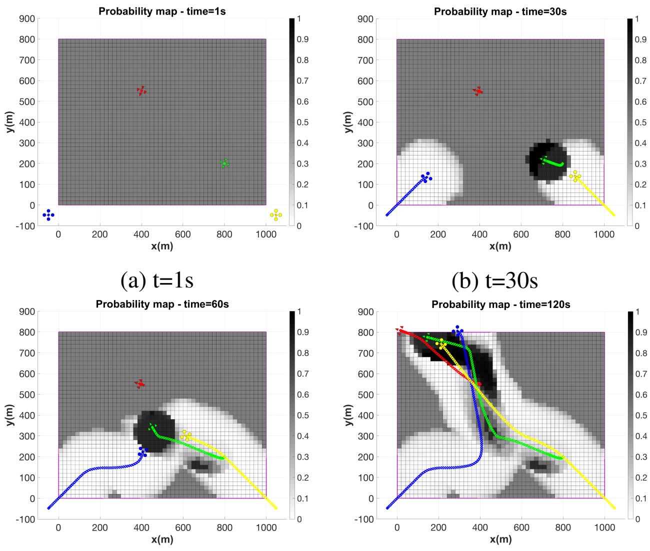
Snapshots of the simulation are included in Figure 8. In Figure 8a, all cells were initialised with $k = 0.5$. At t=30s, in Figure 8b, both UAVs were following a straight line as no birds were detected. The yellow UAV detected the green flock, hence the probability increased in the region. At t=60s, shown in Figure 8c, the blue UAV joined the green UAV to chase the green flock. In Figure 8d, both flocks were successfully deterred as they left the environment. The simulation demonstrated that two flocks of birds on an 8 hectare area can be effectively deterred using the proposed algorithm.

8 Conclusion and Future Research

Bird damage is a very challenging global problem. The solution proposed in this research incorporate bird psychology and autonomous UAVs to overcome the limitation in other methods. A bird deterring strategy and a bird chas-

Table 1: Simulation parameters

Parameter	**Value**	**Parameter**	**Value**
Cell shape	square	Cell size	10×10 m
x_{min}	0 m	$\tau_{interest}$	0.99
x_{max}	1000 m	τ_{prob}	0.99939
y_{min}	0 m	k_{nom}	1
y_{max}	800 m	k_{low}	0.2
UAV max. velocity	10 m/s	k_{high}	0.8
UAV max. yaw rate	45 deg/s	θ_{sen}	120°
Bird max. velocity	8 m/s	r_{unc}	100 m
Bird max. yaw rate	20 deg/s	r_{sen}	200 m



(a) t=1s (b) t=30s

(c) t=60s (d) t=120s

Figure 8: Simulation results

ing algorithm were developed to enable autonomous operation of the proposed system. Simulation results indicated the system had potential in protecting a large vineyard with multiple UAVs. Future research is needed in coordinating multiple UAVs to execute the mission more efficiently. The cost function needs to be adjusted to avoid multiple UAVs chasing after the same flock. Despite the openness of agriculture properties, no-fly region may exist due to high trees or power poles. Future research should modify the cost function to take no-fly region into consideration. Current research is directed towards applying this algorithm on hardware.

ACKNOWLEDGEMENTS

We would like to thank Dr. Andrea Griffin from the University of Newcastle, Darren Fahey from New South Wales Department of Primary Agriculture and Dr. Andrew Lucas from Agent Oriented Software for their support in our manual flight experiment.

REFERENCES

[1] John Tracey, M Bomford, Quentin Hart, Glen Saunders, and Ron Sinclair. *Managing bird damage to fruit and other horticultural crops*. Bureau of Rural Science, Canberra, 2007.

[2] J. L. Elser, A. Anderson, C. A. Lindell, N. Dalsted, A. Bernasek, and S. A. Shwiff. Economic impacts of bird damage and management in U.S. sweet cherry production. *Crop Protection*, 83:9–14, 2016.

[3] Michael R. Conover. *Resolving Human-Wildlife Conflicts*. CRC Press, Boca Raton, FL., 2001.

[4] Sara M. Kross, Jason M. Tylianakis, and Ximena J. Nelson. Effects of Introducing Threatened Falcons into Vineyards on Abundance of Passeriformes and Bird Damage to Grapes. *Conservation Biology*, 26(1):142–149, 2012.

[5] Brian A. Grimm, Brooke A. Lahneman, Peter B. Cathcart, Robert C. Elgin, Greg L. Meshnik, and John P. Parmigiani. Autonomous Unmanned Aerial Vehicle System for Controlling Pest Bird Population in Vineyards. In *Volume 4: Dynamics, Control and Uncertainty, Parts A and B*, page 499. ASME, nov 2012.

[6] Michael R. Conover and James J. Perito. Response of Starlings to Distress Calls and Predator Models Holding Conspecific Prey. *Zeitschrift für Tierpsychologie*, 57(2):163–172, 1981.

[7] Andrea S. Griffin, Hayley M. Boyce, and Geoff R. MacFarlane. Social learning about places: observers may need to detect both social alarm and its cause to learn. *Animal Behaviour*, 79(2):459–465, 2010.

[8] Christopher W Lum, Rolf T Rysdyk, and Anawat Pongpunwattana. Occupancy based map searching using heterogeneous teams of autonomous vehicles. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (August):1–15, 2006.

[9] Zihao Wang, Andrea S. Griffin, Andrew Lucas, and KC Wong. Psychological warfare in vineyard: using drones and bird psychology to control bird damage to wine grape. *Unpublished manuscript*, 2018.

[10] MRobotics. mRo Pixhawk 2.4.6 Cool Kit (Limited edition), url: https://store.mrobotics.io/product-p/mro-pixhawk1-fullkit-mr.htm, 2016.

[11] E. Rosten, R. Porter, and T. Drummond. Faster and Better: A Machine Learning Approach to Corner Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):105–119, jan 2010.

[12] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, pages 246–252. IEEE Comput. Soc, 1999.

[13] Eben Upton. Raspberry Pi 3 Model B, url:https://www.raspberrypi.org/blog/raspberry-pi-3-model-bplus-sale-now-35/, 2018.

[14] Raspberry Pi Foundation. Camera Module, url:https://www.raspberrypi.org/documentation/hardware/camera/README.md, 2016.

Precision Weed Spraying using a Multirotor UAV

Lucas J. de Rijk, Seumas M. Beedie, and Karl A. Stol*
Department of Mechanical Engineering, University of Auckland, New Zealand

ABSTRACT

This paper presents a method for integrating spraying components on a multirotor Unmanned Aerial Vehicle (UAV) in order to perform aerial precision weed spraying. Experimental tests were conducted to assess spray accuracy as a function of tracking dynamics, target position, and UAV motion during manual flights. It was found that high standard deviations of UAV roll and pitch are correlated with poor spray performance and that the implemented system is robust to light wind exposure.

1 INTRODUCTION

Over the past several decades, there has been proliferation and acceptance of emerging technologies in the agricultural industry, particularly relating to Unmanned Aerial Vehicles (UAVs) [1,2]. Improved technology of multirotor UAVs has attracted research on the ability to carry out high precision agriculture-related tasks. One such application is the spraying of weeds, which is of vital importance for crop yields, but can be either time consuming in the case of manual spot-spraying, or expensive and environmentally harmful in the case of boom spraying [3]. While existing UAVs offer blanket spraying of crops, the advantages of these systems over land-based boom spraying are limited. In contrast to ground-based autonomous spraying, an airborne system is faster and not reliant on a traversable surface.

Utilising a visual tracking system enables an autonomous weed spraying platform to both distinguish weeds from surrounding crops and pasture, as well as track the position of a target weed during flight to enable accurate spraying. There is significant previous work [4] relating to the visual identification of weeds from high altitude, but these methods are unsuitable for high speed control scenarios due to high latency and high processing power requirements [5]. Hansen et al. [6] used a medium altitude aerial visual system and global positioning satellite waypoints to direct the movement of a ground-based spraying system, with a Time of Flight (ToF) camera identified as a potential method for identifying weeds at close range, capable of framerates of 30fps, which may be sufficient for controlled tracking.

Investigation into aerial spray systems has been limited, with commercial products utilising wide swathes for crop dusting [7], and academic research focusing on low precision wide coverage spray systems [8] with a root mean square (RMS) error of 0.2m from at an altitude of 5m. In order to precisely spray common herbaceous weeds such as Californian thistles, which have an average diameter of around 110mm [9], a lower altitude and less diffuse spray will be required.

This paper investigates the control system requirements and capabilities for precision weed spraying using a UAV, by first describing the hardware used for testing, followed by an overview of the implemented control system and its characteristics. Finally, the precision of the spraying performance is assessed in flight.

2 EXPERIMENTAL HARDWARE

The experimental system is shown in Figure 1. The UAV is an Aeronavics BOT quadcopter with a motor-to-motor diameter of 1.0 m and flight endurance of 10 minutes when carrying the 1.2 kg spraying system. The Foxtech 3-axis gimbal directs

*contact email: k.stol@auckland.ac.nz

the spray and dynamically isolates the camera and nozzle from UAV attitude changes. The Foxtech gimbal uses an AlexMos 32-bit gimbal controller with two inertial measurement units (IMUs) in order to stabilise the camera and allows inputs to control the gimbal axes.

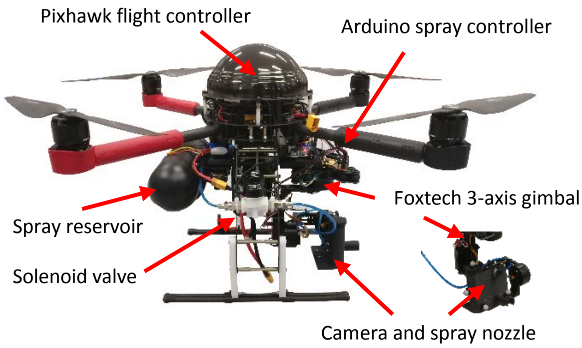


Figure 1 - Multirotor UAV with spray system

For the purposes of this study, accurate weed identification was not necessary, as experiments could be carried out with any visual tracking system with sufficient sampling frequency to allow control goals to be met. The CMUcam5 Pixy is suitable as it allows hue-based visual tracking at 50fps using a resolution of 320x200 pixels and vertical and horizontal fields of view of 47° and 75° respectively. Existing experiments of the Pixy camera implemented for visual tracking on UAVs [10] showed that using the visual input to directly control UAV motion was not an effective method of tracking, resulting in large overshoot and slow transient response of 2-4 seconds. These results support the use of a pan/tilt gimbal for the purposes of visual tracking, which have been used to achieve errors of less than 10 pixels [11].

In place of a weed, bright red targets printed onto paper were used to provide a high contrast against the laboratory floor background, allowing consistent visual identification as well as enabling visual analysis of the result of each test to quantify spray performance. While some phenomena such as liquid splashing and spreading over the ground are not emulated correctly using a flat target, this experimental setup does provide a useful measurement of spray accuracy.

An Arduino Mega 2560 was used to implement a controller that utilizes feedback from the Pixy camera and provides output to the gimbal controller and solenoid spray valve. The Arduino was also used to receive supervisory input from a smartphone over Bluetooth.

3 CONTROL SYSTEM IMPLEMENTATION

A spherical coordinate system, Figure 3, is used to describe the target location relative to the camera frame. The horizontal pixel error measured by the camera is used to measure azimuth angle and the vertical pixel error measured by the camera is used to measure elevation angle.

Figure 3 illustrates the controller for the gimbal yaw axis, for regulating azimuth angle Ψ. The controller for the gimbal pitch axis, to regulate the elevation angle ϑ, is identical in structure, whereas the roll controller does not have a visual feedback loop because a change in the gimbal roll axis does not result in a change in spray direction.

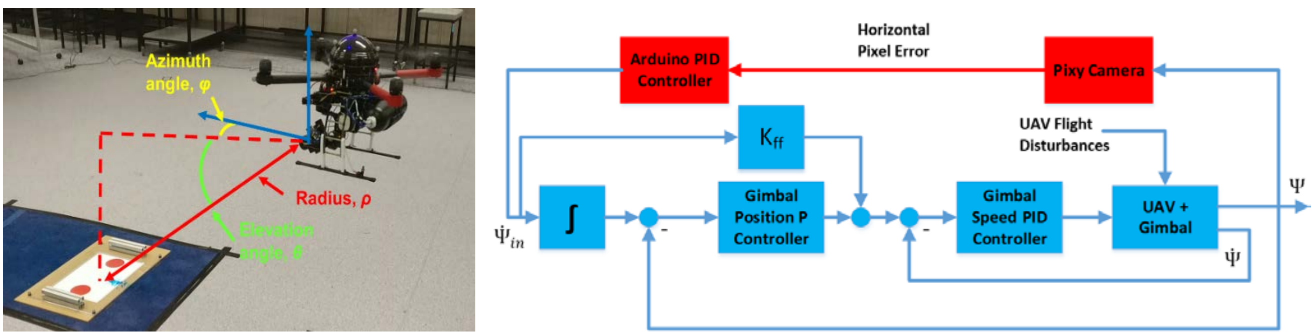


Figure 3 - Target coordinates and control system block diagram for azimuth tracking

Shown in blue in Figure 4 is the gimbal control system using cascade position and speed controllers. It allows a user to give a speed control input, which correspondingly adjusts the position setpoint at which the gimbal is stabilised, as well as giving a direct speed adjustment using a feedforward gain. Shown in red in Figure 4 is the added vision feedback loop, which provides a velocity control signal to the gimbal calculated using a PID controller based on the pixel error between the centre of the frame and the target centroid. PID gains and sampling frequency are adjustable through the alteration of the Arduino program.

The implemented PID controllers were tuned heuristically, with the aim of improving the disturbance rejection of the system as much as possible, without introducing oscillation in response to a large step input which occurs when a target is introduced at the edge of the image frame.

4 SYSTEM CHARACTERIZATION

4.1 Closed-Loop Frequency Response

The closed-loop frequency response of the tracking system was determined through the use of a VICON motion capture system. Motion capture markers were attached to the camera and target, and the UAV was moved by hand for two minutes with the tracking system operating, with the goal of providing a wide range of excitation frequencies.

Transfer functions were then approximated for each of the two relevant directions using the input data (the azimuth and elevation angles) and the output data (the yaw and pitch angles of the camera respectively). These transfer functions were used to determine the closed-loop frequency response, Figure 4. The corner frequency is 0.78Hz for the yaw axis, and 1.48Hz for the pitch axis. The difference in corner frequency is likely due to the system having greater rotational inertia about the yaw axis.

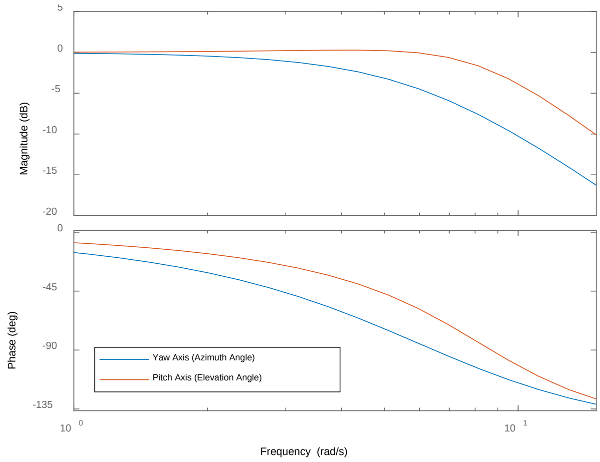


Figure 4 - Bode plot for gimbal yaw and pitch

A spectral analysis using the Fast Fourier Transform was carried out on the azimuth and elevation angle data collected during later flight tests (see Section 5) for the two flights with the highest UAV roll and pitch standard deviation respectively. It was found that the majority of excitation angle change occurred at frequencies of less than 0.5Hz as shown in Figure 5. However, as spray performance did appear to be negatively impacted at these levels of roll and pitch standard deviation, it is unlikely that the corner frequency is an accurate measure of the excitation frequency below which spray performance will not be negatively affected.

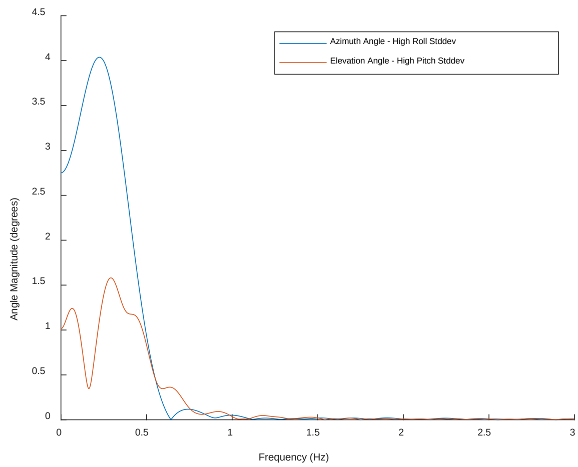


Figure 5 - Frequency spectrum of flight test data

4.2 Effect of Sampling Frequency

A study was carried out to determine how corner frequency degrades with reduced sampling frequency. This was achieved by sampling the camera at reduced frequencies, and again determining the closed-loop frequency response as in section 4.1. To reduce the number of variables changing between tests, excitation was only provided in the yaw direction and the target was moved instead of the UAV to avoid the gimbal needing to correct for changes in UAV orientation. To gain a more accurate estimate of the corner frequency at each sampling frequency, transfer functions of up to fifth order were fitted to the data, and the convergence of these fitted functions, Figure 6, was used to establish the actual corner frequencies. The data does support the overall expected trend that increasing sampling frequency results in a higher corner frequency, but there is uncertainty present in the graph such as the unexplained local minimum at 40Hz sampling frequency. This may be due to the limited duration of the tests, as well as non-uniform excitation signals between trials which may have resulted in less data at some frequencies. Corner frequency does not seem to be significantly affected until below sampling frequencies of 30Hz. For comparison, template matching techniques have been able to achieve sampling frequencies between 26Hz and 29Hz using a small form factor PC/104+ for processing [12].

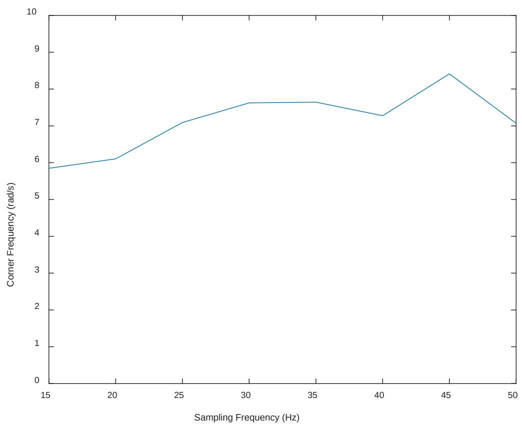


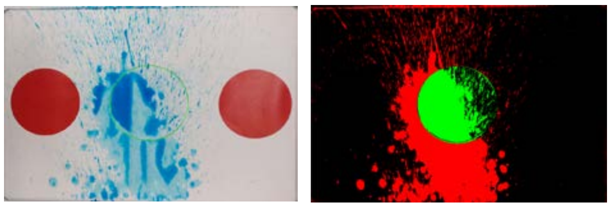
Figure 6 – Effect of sampling frequency

4.4 Spray Duration

A study was carried out to determine the ideal duration of the spray in order to provide high coverage of the target without unnecessary spillage beyond the edge of the target. This spray duration refers to the total amount of time for which the solenoid valve is open. If the pixel error exceeds a 10-pixel threshold while spraying, the spray will be interrupted, and will only resume when the pixel error drops back within the acceptable range. It was found that any spray duration exceeding 1.1 seconds would increase spillage without increasing coverage, so a 1.1 second spray duration was used for all flight tests. In all tests the fluid reservoir was pressurised to 100kPa, corresponding to a spray velocity of 7.5m/s at the nozzle.

5 FLIGHT TESTS

Spray performance was measured by using blue dyed water as a spray liquid, in conjunction with a computer vision analysis. Rather than spraying a coloured target directly, the mean position of two red targets on a sheet of paper was tracked, with the target spray area being a 110mm diameter circular outline between the two red targets, Figure 7(a). Photos were taken of the result of each spray test and an OpenCV program was written to measure the *coverage* (expressed as a fraction of the target circle covered with spray) and the *spillage* (area outside the circle covered with spray, also expressed as a fraction of the target area). Computer analysis of a typical spray test showing coverage in green and spillage in red can be seen below in Figure 7(b). This particular test had a coverage of 0.76 and a spillage of 2.06.



(a) (b)

Figure 7 - Spray test sample: (a) raw image (b) image analysed in OpenCV

5.1 Effect of UAV Movement

A total of 26 independent flight tests were carried out using a human pilot rather than an automated flight controller to provide with a wide range of UAV motion patterns. Variables considered in the analysis were the mean and standard deviation of range, elevation angle, and azimuth angle, as well as the standard deviation of roll, pitch, and yaw. A statistical analysis was undertaken to find which variables had the most significant effect on coverage and spillage.

The strongest relationship encountered for coverage involves UAV pitch standard deviation. A linear fit to this data, Figure 8, explains 44% of variation in the data. This trend likely exists because UAV pitch causes both the range and elevation angle to change. High pitch amplitudes result in higher amplitude and frequency changes in range and elevation angle, which the tracking system cannot reject effectively.

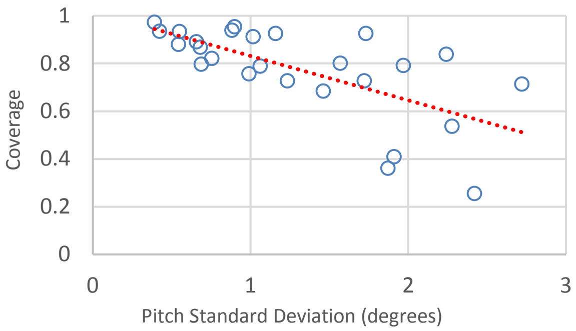


Figure 8 – Coverage vs UAV pitch variation

The second strongest relationship encountered for coverage was a negative relationship between UAV roll standard deviation and coverage, due to the gimbal being located below the UAV centre of mass.

To give an idea of the significance of these relationships, it was found that for the 12 tests where both roll and pitch standard deviation were less than 1°, the mean coverage was 87.7% of the target, compared to a mean coverage of 77.6% across all 26 flight tests.

There was no significant relationship between spillage and any of the measured motion variables. There was also no significant relationship found between target range and coverage, Figure 9, up to the maximum tested range of 2.2m.

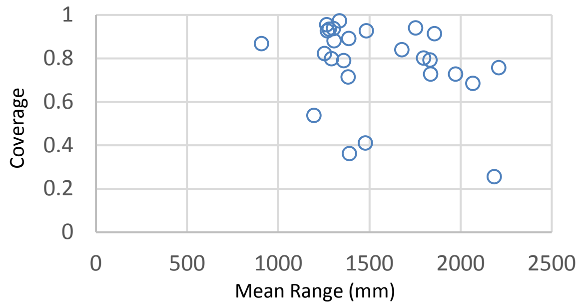


Figure 9 - Coverage vs target range

5.2 Wind Disturbance

Turbulent wind was introduced using two 540mm diameter fans, positioned such that they would influence both the UAV and the spray. A perpendicular cross-wind was tested as it was deemed the most likely to negatively influence the spray performance. The windspeed was measured using an anemometer to be an average of 4.0m/s at the UAV position, 2.2m/s across the spray path between the centre of the two fans, and 2.0m/s across the surface of the target, corresponding to a light or gentle breeze on the Beaufort scale.

Two flights were conducted, resulting in a high coverage (91.2% and 97.6%) and a typical amount of spillage compared to the other tests (208.8% and 267.3%) as seen in Figure 10. Each data point represents an individual test. This indicates that the light wind exposure had little to no impact on the effectiveness of the system, despite high UAV roll standard deviations of 1.33° and 1.44° compared to a mean of 0.83° for the tests without wind exposure.

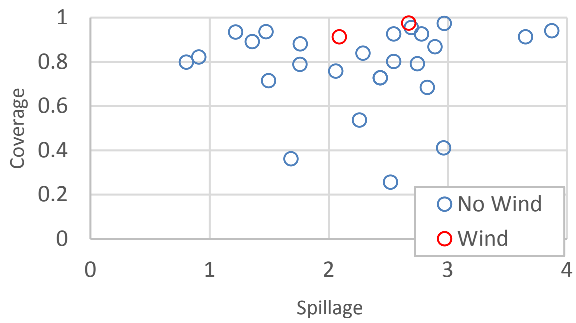


Figure 10 - Impact of light wind disturbances

6 CONCLUSIONS

An aerial precision spraying system has been developed utilising a gimballed sprayer and vision-based feedback for target tracking. Hue based tracking with a CMUcam5 Pixy simplified the vision system, and as such allowed a focus on the control requirements. The system was characterised using a range of tests and it was determined that the corner frequencies (baseband bandwidths) were 0.78Hz and 1.48Hz for the yaw and pitch axes respectively. It was also determined that these bandwidths are insensitive to sampling frequencies (vision frame rates) down to 30Hz.

Under controlled indoor conditions it was found that high UAV pitch and roll standard deviations were related to decreases in spray coverage. Flight tests with less than 1° of standard deviation of the UAV pitch and roll angles had a target coverage of more than 10 percentage points higher than the overall average coverage. Finally, two flight tests were carried out under exposure to light wind disturbance and it was found that this had little effect on the spray performance.

Future work could include the testing of the system outdoors using suitable station keeping controllers and the further development of weed tracking and identification from literature.

REFERENCES

[1] M. J., Grassi, “Rise of the Ag drones.” http://www.precisionag.com/specialty-crops/rise-of-the-ag-drones/, Mar. 6, 2013 [Apr. 29, 2017].

[2] K. Sayler, “A world of proliferated drones: a technology primer,” Center for a New American Security, 2015. https://www.files.ethz.ch/isn/191911/

[3] W. Zhang, F. Jiang, and J. Ou, “Global pesticide consumption and pollution: with China as a focus,” in Proc. of the Int. Acad. of Ecology and Env. Studies, 2011.

[4] A. Collings. “Drones could bring benefits to arable farmers.” http://www.fwi.co.uk/machinery/drones-could-bring-benefits-to-arable-farmers.htm, Feb. 1, 2013 [April. 27, 2017].

[5] J. Blasco, “AE – Automation and emerging technologies: Robotics weed control using machine vision,” Biosystems Engineering, 83(2), pp. 149 – 157, 2002.

[6] K. D., Hansen, F. Garcia-Ruiz, W. Kazmi, M. Bisgaard, A. la Cour-Harbo, J. Rasmussen, and H. J. Andersen. “An autonomous robotic system for mapping weeds in fields,” in 8th IFAC Symp. on Intelligent Autonomous Vehicles, 2013.

[7] A. Sato, “Report: The RMAX helicopter UAV,” Aeronautic Operations, Yamaha Motor Co Ltd. Sept. 2, 2003.

[8] X. Xue, Y. Lan, Z. Sun, C. Chang, and W. C. Hoffman, “Develop an unmanned aerial vehicle based automatic aerial spraying system,” Computers and Electronics in Agriculture, 128, pp. 58 – 86, 2016.

[9] G. Blackwell, D. Lucock, H. Moller, R. Hill, J. Manhire, and M. Emanelsson, “Abundance and diversity of herbaceuous weeds in sheep/beef pastures, South Island, New Zealand,” NZ J. of Agricultural Research, 54(1), pp. 53 – 69, 2011.

[10] T. X., Dao, “Video-guided autonomous pollinator rotorcraft,” Master’s thesis, Dept. Aero. Eng., San Jose State Univ., California, 2013.

[11] X. He., and S. Yi-ping, “Target tracking control of mobile robot in diversified manoeuvre modes with a low cost embedded vision system,” Industrial Robot, 40(3), pp. 275 – 287, 2013.

[12] Qadir, W. Semke., and J. Neubert, “Implementation of an onboard visual tracking system with small unmanned aerial vehicle (UAV),” Int. J. of Innovative Technology and Creative Engineering, 1(10), pp. 17 - 25, 2012.

Visual Servo Control in Quadrotor for Power Line Tracking

Luciano do V. Ribeiro*[1], Caique de S. F. Duarte[1], Alexandre C. B. Ramos[1], and Félix Mora-Camino[2]

[1]Federal University of Itajubá, Itajubá, MG, Brazil
[2]Federal University Fluminense, Niterói, RJ, Brazil

Abstract

In this work, Visual Servo Control is employed for the guidance of an Unmanned Aerial Vehicle quadrotor over the transmission power lines during an inspection. With this technique, it is possible to fly the UAV in a GPS-denied environment. A camera is attached in the UAV and with the processing of the images, the lines forming the transmission lines are extracted. These lines are used to feed a PID controller to correct the UAV pose. The image processing is made in a Raspberry Pi 2 embedded in the UAV and the software was made using the ROS middleware.

1 Introduction

Electric energy distribution companies carry out regular inspections and preventive maintenance to ensure the operation of the network. With the inspections, it is possible to analyze the health of the components that are part of the transmission line like the conductors, insulators, tower and the vegetation near the power line corridor. Traditionally the surveys are made by a manned helicopter containing sensors to data acquisition, as high-resolution cameras. Such data will be analyzed later for identification of anomalies. This type of work is very dangerous, costly and time consuming.

Different approaches to conducting inspections in transmission lines have been studied [1]. In recent years, interest in the use of VTOL (Vertical Takeoff and Landing) Unmanned Aerial Vehicles (UAVs) to automate the acquisition of images for transmission line inspection has grown [2]. Inspection with UAV platforms can be made remotely controlled by an experienced pilot, or autonomously, with the pre-programmed route in the form of waypoints in the Ground Control Station (GCS).

UAVs typically rely on GPS signal fused with the IMU (Inertial Measurement Unit) data to calculate your position and speed over time. However, its accuracy is not good enough to allow a flight near structures as is the case of line tracking. Another method for determining its position should be investigated. Computational visual and image processing techniques can be used to obtaining the position and attitude of a UAV.

*Email address: lucianoribeiro@unifei.edu.br

In this work, a low cost VTOL UAV platform was developed for inspections on transmission lines. The main objective of the project is to develop an autonomous navigation system to make the path between two consecutive towers. Visual information will be used to correct the orientation and position of the UAV along the path. The UAV will run over the transmission line, and a downward-facing camera will capture images. An embedded computer will identify the transmission lines and errors for the orientation and distance perpendicular to them will be determined. Finally, a PID controller will be implemented to correct the route, using image processing as feedback.

The remainder of this paper is organised as follows: In section 2 are presented work related to inspection of transmission lines with UAVs. In Section 3, the UAV platform developed and the problem description will be performed. The architecture of the image processing software and simulations are presented in Section 4. And finally, the conclusion in section 5.

2 Background and Related Work

The researches for inspection of transmission lines are usually divided into two groups. The first is focused on developing techniques for identifying lines in the image and another on developing control techniques given the line that must be followed. For the identification and tracking of transmission lines, some considerations are usually made:

- Transmission lines are linear objects viewed from above.
- They have a uniform brightness due to the material they are made of.
- The power lines are parallel to each other.
- They occupy most of the image.

Given these characteristics, most algorithms for detecting transmission lines are based on the Hough Transform and some other technique for grouping the detected lines, to discriminate lines that are not transmission lines. The great difficulty is in discriminating the transmission lines of other linear objects from the background.

Zhengrong Li et al. [3, 4] had developed a PCNN (Pulse-Coupled Neural Network) filter with the objective of removing background noise in the detection stage of the lines. Then Hough Transform is applied for straight line detection and K-means is used to cluster straight lines that have the same slope. Their implementation was tested on real data captured by aircraft but was not used for guidance.

Yang et al. also used the Hough Transform to detect lines in the binarized image [5]. As a clustering method, fuzzy C-means was implemented, having as features to determine a cluster the size and slope of the lines.

Mills e Aouf [6] had developed an Imaged Based Visual Servo Control application for orientation correction and displacement of a fixed-wing aircraft, to follow the transmission line in the presence of wind. The results were obtained in simulation.

Zhang et al. made an important work in the detection and tracking of transmission lines to be used in real time [7]. In addition to Hough Transform and K-means for straight line clustering, the Kalman filter is applied to track the straight lines in the Hough space. The processing time of each frame was 40ms, ideal to be realized in real time, but the hardware used was not described, and if it has the capacity to be embedded in a UAV

Araar e Nabil [8] had developed two solutions, IBVS e PPBVS. Their proposal was tested in a simulated and real environment. Not much focus was given on transmission line detection and tracking. It was not considered a complex background for the power lines. However, the way the detection was handled was used as a reference in our solution.

3 Materials and Methods

3.1 UAV Plataform

A low-cost 450mm frame quadrotor was developed for this project, as can be seen in Figure 2. The quadrotor is equipped with four 1000KV brushless motors and 8 × 4.5" propellers. The open-hardware Pixhawk autopilot board was used [9]. It is responsible for the low-level control of the quadrotor. It is equipped with accelerometers, gyroscopes, magnetometers and with a GPS + compass mounted externally. The firmware running on Pixhawk is the ArduPilot stack [10]. It is responsible for reading the sensors, the radio controller signals and independently controlling each motor. It also implements MAVLink, a protocol developed especially for UAVs that allows the communication with other devices.

A Raspberry Pi 2 was embedded in the quadrotor to enable autonomous flight. It has Ubuntu-Mate 16.04 and the middleware Robot Operating System (ROS) Kinetic installed [11]. A downward-looking Raspberry Pi camera was attached to the quadrotor and images with 640 × 360 at 30 fps are sent to the onboard computer. Through image processing, our software will calculate the correction velocities (angular and linear) and send them to the Pixhawk through the serial port using the MAVLink protocol.

A pilot with Radio Controller will also be able to take control of the quadrotor at any time. The quadrotor has a long-range radio 900Mhz that communicates with the GCS (Ground Control Station), which is possible to configure some parameters and get real-time telemetry data. The laptop running the GCS also has ROS, which makes possible to control and debug our software that runs on the embedded computer.

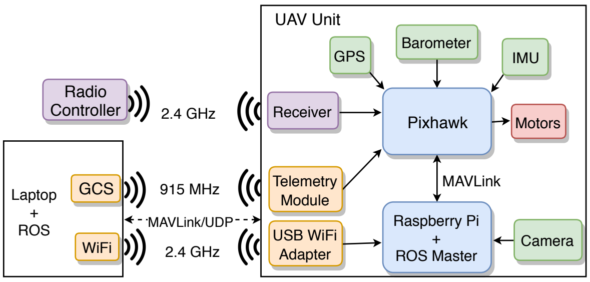


Figure 1: Hardware architecture, adapted from [12]

Figure 2: Developed UAV quadrotor

3.2 Mission Description

One of the main tasks in the inspection of transmission lines is the capture of images of the network components that are in the tower. In order to enable autonomous data capture, it is necessary for the UAV to be able to move between the towers for the execution of its mission. While transmission companies have data on the geographic coordinates of the towers, this data is for reference only and not its exact position. Além disso, o próprio GPS pode conter imprecisões e até mesmo ficar indisponível em algumas cituações. Therefore, making the flight considering only GPS data is not enough. In this project, visual information of the transmission lines will be used as feedback to control the route of the UAV in the path between the towers.

For the beginning of the mission, the UAV will start above the initial tower and go through an initialization phase, to identify the transmission lines. The objective of the UAV will be to carry out a route that follows the lines, keeping its orientation parallel with the transmission line and minimizing the perpendicular distance between the UAV and the transmission line. Figure 3 illustrates this situation. On the left, we have the representation of the UAV and its field of view and the right, the orientation gap between the UAV heading and the center line, and the perpendicular distance between the UAV and the center line.

Our control strategy does not take into account the current position and orientation of the UAV, only its position regarding the transmission line, which can be identified by the image processing. However, as will be seen in Section 2, the output of our program are linear and angular velocity commands, which are sent to the flight controller via MAVLink. In order for Pixhawk to implement these commands, it needs to have UAV speed information, so we will use the GPS signal. Other techniques for obtaining the current speed can be used, such as the image processing itself. There is hardware that can be integrated with Pixhawk that performs this type of processing, and it could also be done on the embedded computer, but this is out of the scope of our project.

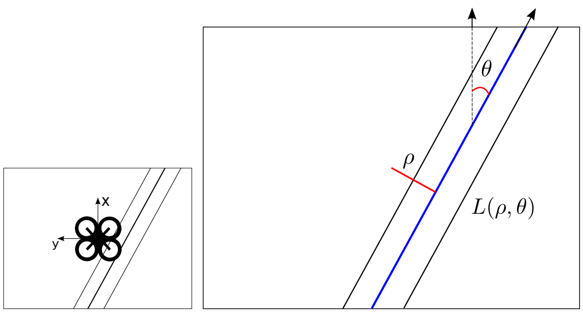


Figure 3: Representation of the image plane and the features extracted from the power lines

4 Results and Discussion

4.1 System Architeture

This project was developed using the ROS middleware and programmed in C++. With ROS it is possible to create a distributed network of processes, called nodes, that can communicate with each other through messages. The concept of publisher and subscriber is used. Each node informs which topics it wants to publish data to and which it wants to subscribe. Each topic has a name and an associated data type. There is a Master node that allows nodes to discover others on the network.

Our architecture consists of four nodes. A node is responsible for being the camera driver and sending images to the node that will handle the processing. The Image Processing Node will identify the transmission lines from the background and will get the center line.

This line will be sent to the main node responsible for the control of the position and heading of the UAV. In the main loop, the error of rotation and translation will be used as feedback for a PID controller. Speed commands will be sent to Pixhawk through mavros, a ROS node that implements the MAVLink protocol. mavros node is also responsible for converting between the coordinate system used by ROS and Pixhawk.

On the laptop running the GCS it is possible to have a ROS client. With it you can receive information about the nodes that are running on Raspberry Pi and also send some command, such as starting or stopping the execution of the mission.

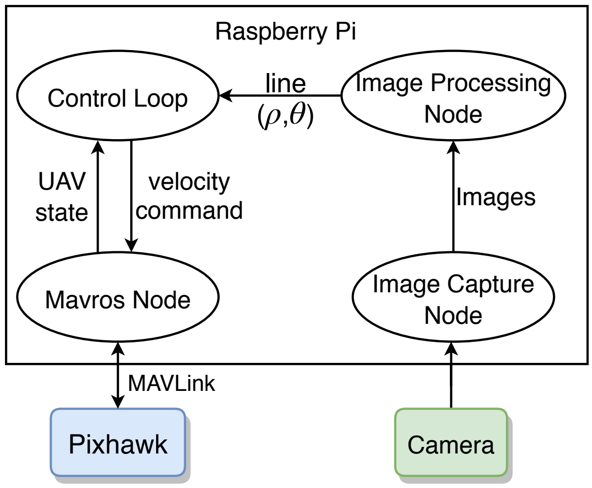


Figure 4: System architecture

4.2 Algorithm Implementation

4.2.1 Image Processing

OpenCV will be used for image processing. This is an open-source library with implementation of several algorithms for computer vision.

Each frame that arrives from the camera goes through a pre-processing. The image is converted to grayscale and then a media filter is applied for noise reduction. Then Canny filter is applied. It is responsible for identifying the edges in the image. This is a fundamental process because it reduces the amount of information present in the image and the processing time of the subsequent steps. The output of the Canny algorithm is a binary image, where the white pixels correspond to the edges of the image. Standard Hough Transform will be applied over this image to identify the straight lines.

With HT it is possible to identify any shape (line, circle, ellipse) that can be expressed analytically. In the case of straight lines, each point (x, y) in the image plane that belongs to a border will be mapped to a curve in the Hough space according to the Equation 1.

$$x \cos\theta + y \sin\theta = \rho. \tag{1}$$

For each point, all values of (ρ, θ) that satisfy the Equation 1 will have their frequency increased in Hough space. The parameters (ρ, θ) that has the highest amount of votes will be the corresponding to the line with more collinear points.

Just as it was done in [8], the HT output implemented by OpenCV needs to be modified to better serve our purpose. The parameter ρ, which represents the distance from the origin of the image (upper left corner) to the straight line, will become the distance from the center of the image to the line, using

$$\rho = \rho - \frac{W\cos\theta + H\sin\theta}{2} \tag{2}$$

where W e H represent the width and height of the image respectively. For the case of the center line, the value of ρ will be the error of the position of the quadrotor relative to the center of the image. The parameter θ will be in range $[0, \pi]$ and the desired setpoint for the UAV is to have θ fixed on zero. As there will be oscillations, around the point 0, we will have discontinuity and this will lead to an unstable control. The solution is to translate the value of θ to the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$ according to

$$If \;\; (\theta \geq \frac{\pi}{2}) \quad Then: \begin{cases} \theta = \theta - \pi \\ \rho = -\rho \end{cases} \tag{3}$$

After extracting all the lines that are present in the image, it is necessary to identify those that are part of a transmission line. Our implementation takes into account the existence of a predetermined number of lines to be identified, which in our case is 3. Since the image does not change much between two consecutive frames, the line that has the most votes in HT and is close to the chosen line in the last frame will be chosen for the current frame.

$$\begin{cases} |\rho - \rho_{last}| < \rho_{max} \\ \quad\quad \text{and} \\ |\theta - \theta_{last}| < \theta_{max} \end{cases} \tag{4}$$

.

The other two lines are chosen in such a way that they are parallel to the first, $|\theta_i - \theta_1| < \theta_{max}$, and that does not refer to the same edge, $|\rho_i - \rho_1| > \rho_{min}$.

4.2.2 PID controller

Our algorithm will control the movement of the UAV only in the *2D* plane, not changing its height. For this two PID controllers will be implemented. One to correct the guideline of the quadrotor and another to correct its deviation perpendicular to the line. The mathematical expression of a PID controller is given by Equation 5

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{5}$$

where a command $u(t)$ will be proportional to the error $e(t)$, the sum of this error over time and the rate of change of this error in the instant t. As can be seen in Figure 3, the error in the direction and position of the UAV in relation to the center line are the values θ, ρ, respectively. As presented in [8], the error in ρ tends to increase as the UAV moves in the direction that should decrease. This is due to the fact that the frame has to tilt to perform the position correction. This tilt moves the line in the plane of the image in the opposite direction, giving the impression that the distance has increased. We will use a gimbal to stabilize the field of view of the camera parallel to the ground in these situations.

4.3 Simulation

Simulations to test the ability to identify straight lines and mainly the control algorithm was made in the simulator Gazebo [13]. With the Gazebo simulator, it is possible to create different environments, simulate sensors and actuators. Within the developer community, you can find numerous sensors and actuator types to use on robots.

A simple simulation environment has been developed, as can be seen in Figure 5.

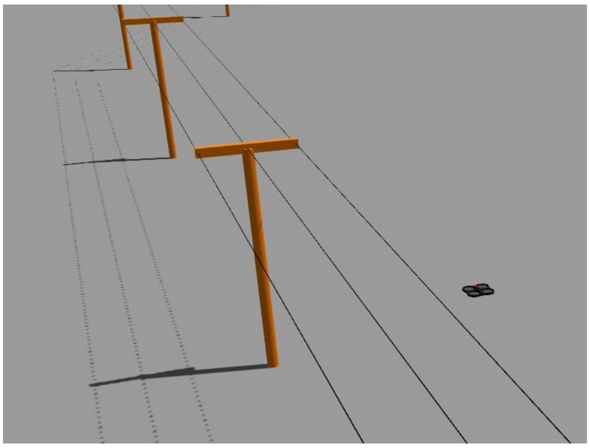


Figure 5: Simulated environment in Gazebo

In the Figure 6 you can see two steps of image processing performed on the same frame. At the top we have the image obtained after the application of the Canny filter and below the identification of the main line with the HT.

5 Conclusion

This paper has presented a Visual Servoing Control algorithm for power line inspection using a low cost quadrotor

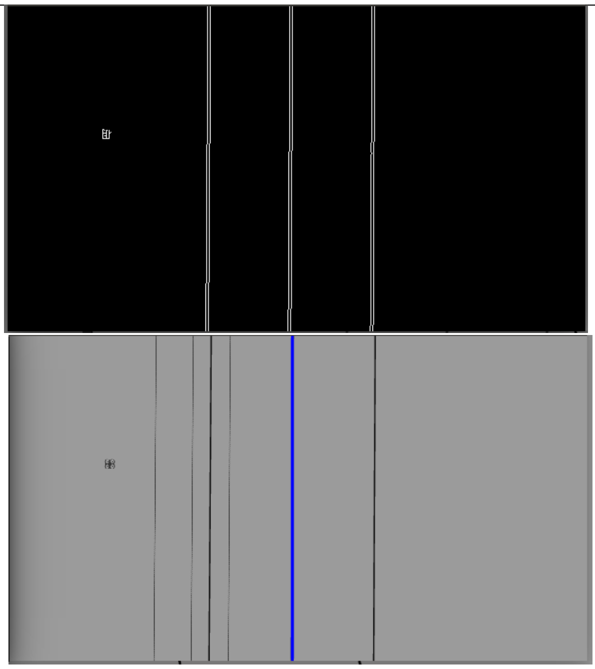


Figure 6: Edge map and Hough Transform applied in the same frame

UAV. The proposed method allows to fly with a quadrotor over the transmission line and to use visual feedback through the image processing to correct the trajectory. The project is being implemented using the ROS middleware and will be embedded on a Raspberry Pi 2.

Currently, testing is being done on the Gazebo robot simulator. The implementation is still simple and further improvements are needed until practical tests are performed.

In addition to making the simulation environment more realistic, we need to make the line detection more robust by using information from previous frames to reduce processing time and increase the accuracy of this step.

Acknowledgements

The authors would like to thank the funding institution: CAPES.

References

[1] Leena Matikainen, Matti Lehtomäki, Eero Ahokas, Juha Hyyppä, Mika Karjalainen, Anttoni Jaakkola, Antero Kukko, and Tero Heinonen. Remote sensing methods for power line corridor surveys. *ISPRS Journal of Photogrammetry and Remote Sensing*, 119:10–31, 2016.

[2] A Pagnano, M Höpf, and R Teti. A roadmap for automated power line inspection. Maintenance and repair. *Procedia CIRP*, 12:234–239, 2013.

[3] Zhengrong Li, Yuee Liu, Ross Hayward, Jinglan Zhang, and Jinhai Cai. Knowledge-based power line detection for UAV surveillance and inspection systems. In *2008 23rd International Conference Image and Vision Computing New Zealand, IVCNZ*, 2008.

[4] Zhengrong Li, Yuee Liu, Rodney Walker, Ross Hayward, and Jinglan Zhang. Towards automatic power line detection for a UAV surveillance system using pulse coupled neural filter and an improved Hough transform. *Machine Vision and Applications*, 2010.

[5] Tang Wen Yang, Hang Yin, Qiu Qi Ruan, Jian Da Han, Jun Tong Qi, Qing Yong, Zi Tong Wang, and Zeng Qi Sun. Overhead power line detection from UAV video images. *International Conference on Mechatronics and Machine Vision in Practice*, pages 28–30, 2012.

[6] Steven Mills, Nabil Aouf, and Luis Mejias. Image based visual servo control for fixed wing UAVs tracking linear infrastructure in wind. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2013.

[7] Jingjing Zhang, Liang Liu, Binhai Wang, Xiguang Chen, Qian Wang, and Tianru Zheng. High speed automatic power line detection and tracking for a UAV-based inspection. *Proceedings of the 2012 International Conference on Industrial Control and Electronics Engineering, ICICEE 2012*, pages 266–269, 2012.

[8] Oualid Araar and Nabil Aouf. Visual servoing of a quadrotor UAV for autonomous power lines inspection. *22nd Mediterranean Conference on Control and Automation*, pages 1418–1424, 2014.

[9] Pixhawk homepage: https://pixhawk.org/. Acessed: August, 2018.

[10] Ardupilot homepage: http://ardupilot.org/. Acessed: August, 2018.

[11] ROS homepage: http://www.ros.org/. Acessed: August, 2018.

[12] Rafael G. Braga, Roberto C. da Silva, Alexandre C.B. Ramos, and Felix Mora-Camino. Collision avoidance based on reynolds rules: A case study using quadrotors. *Advances in Intelligent Systems and Computing*, 558:773–780, 2018.

[13] Gazebo homepage: http://gazebosim.org/. Acessed: August, 2018.

MAV payload: An air-quality monitoring system for integration inside a drone

Christina Andrikopoulou[1*], Stéphane Le Calvé[1,2], Christophe Sutter[1], Florian Noël[2], Alberto Rodríguez-Cuevas[2], Irene Lara-Ibeas[1], Renaud Kiefer[3], Mark Verdines[3], Thomas Pavot[3], Martin Lefebvre[3], Jörg Ettrich[4], Werner Schröder[4], Stefan Junk[4], Steffen Schrock[4]

[1] University of Strasbourg, Institute of Chemistry and Processes for Energy, Environment and Health (ICPEES), 25 rue Becquerel, 67087, Strasbourg France.
[2] In'Air Solutions, 25 rue Becquerel, 67087 Strasbourg, France.
[3] INSA Strasbourg, 24 Boulevard de la Victoire, 67000 Strasbourg, France.
[4] Institute for Unmanned Aerial Systems, Hochschule Offenburg University, Badstraße 24, 77652 Offenburg, Germany.

ABSTRACT

Outdoor air pollution attracts great interest due to its influence on the environment and on human health. To respond to the necessity of outdoor air monitoring, this work presents the conception and development of a sensors-based air monitoring system that meets all the specifications to be integrated inside a drone. Sensors have been chosen to monitor Volatile Organic Compounds (VOCs), Nitrogen dioxide (NO_2) and Ozone (O_3) as well as Temperature (T), Relative Humidity (RH) and Pressure (P). A microfluidic chip consisting of a narrow central microchannel and two wider external microchannels will be added to the system, thus ensuring a satisfactory flow restriction as demonstrated by flow simulations (22% of the initial flow rate).The total monitoring system wil occupy a place of 250 x 170 x 105 mm^3 and its consumption will not exceed 10 W. The latter will be covered by a battery rendering the system autonomous. Two VOCs sesnors have been tested and calibrated by injections of BTEX (Benzene, Toluene, Ethylbenzene and Xylenes) and the results demonstrate very good linearity of the signal as function of BTEX concentration.

1 INTRODUCTION

Outdoor air pollution is a major environmental risk influencing world population's health and life quality. In cities as well as rural areas, it was estimated to cause 3 million premature deaths worldwide in 2012 [1]. Among the various compounds present in outdoor air, Volatile Organic Compounds (VOCs), Nitrogen Oxides (NOx) and ground level Ozone (O_3) play a crucial role in air pollution. More specifically, VOCs and NOx react in presence of light resulting in photo-oxidation products such as O_3, NO_2, PeroxyAcyl Nitrates (PANs) and aldehydes [2]. A series of reactions including the latter compounds and their precursors are responsible for the photochemical smog [2] that many cities experience nowadays. The 2005 "WHO (World Health Organization) Air quality guidelines" offer global guidance on thresholds and limits for key air pollutants that pose health risks. For instance, in the case of O_3 a 100 µg/m3 8-hour mean value has been established, while for NO_2 a 40 µg/m3 annual mean and a 200 µg/m3 1-hour mean have been set. However, in 2014, 92% of the world population was living in places where the WHO air quality guidelines levels were not met [1]. These observations highlight the importance of outdoor air monitoring which is currently mandatory in European countries [3]. For this purpose, air quality monitoring ground stations are used. In parallel, to better respond to the need for outdoor air pollution monitoring and mapping, the integration of monitoring systems in drones that can fly in the range of the troposphere seems

*contact email: candrikopoulou@unistra.fr

to be very promising. Such a solution is proposed in this work, as part of the ELCOD (Endurance LOw Cost Drone) Project.

2 CONCEPTION OF THE MONITORING SYSTEM

One of the two different proposed designs for the drone are presented in Figure 1. The drone will be powered by fuel cells to increase the flight range and to avoid emissions interfering with the sensors' measurements. The payload refers to the sensors-based monitoring system. Important constraints had to be met for the integration of the later in a drone, such as low weight, limited dimensions as well as autonomy and low energy consumption, all balanced with the major necessity for accurate monitoring. Therefore, a microfluidic system based on industrial sensors is suggested in this work.

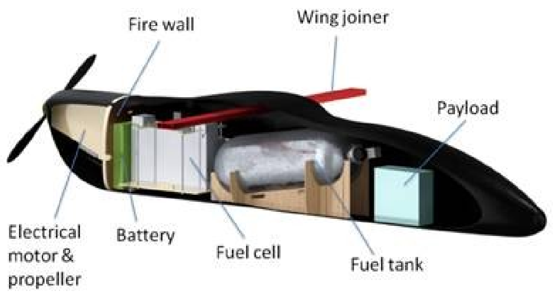
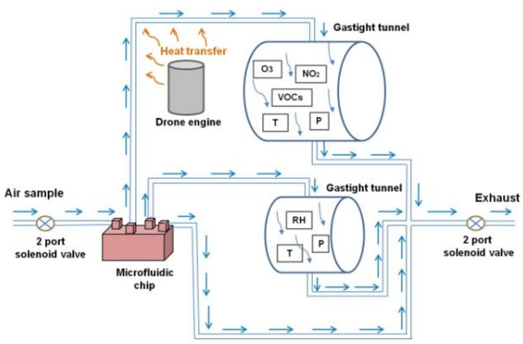


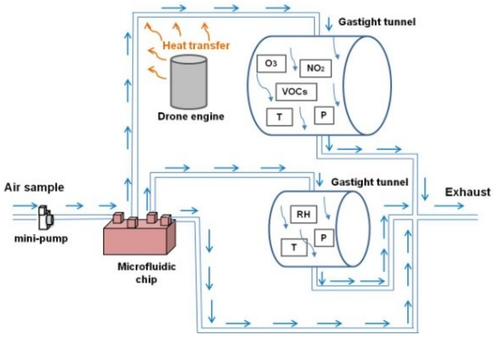
Figure 1 – One of the proposed drone designs (Payload refers to the sensors-based monitoring system).

Chemical sensors meeting the desired performance and characteristics have been chosen for monitoring of VOCs (2 different sensors), NO_2 and O_3. Sensors for Temperature (T), Relative Humidity (RH) and Pressure (P) are also proposed to measure meteorological conditions.

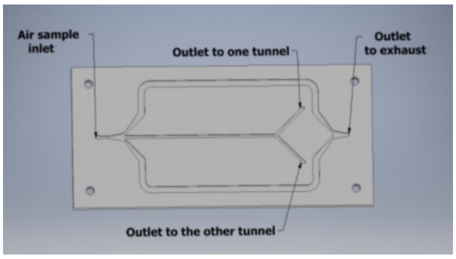
For accurate and reliable measurements during flight, the sensors will be confined inside two gastight cylindrical tunnels (Figure 2a and 2b). One tunnel will contain the chemical sensors and the other the sensors of meteorological conditions. In the tunnel of chemical sensors a T and a P sensor will be also included for calibration purposes. Two different system configurations are proposed to enable sampling during motion and stationary sampling, where 2-port solenoid valves (Figure 2a) and a mini-pump (Figure 2b) will be integrated, respectively. Environmental conditions can have an important influence on the sensors performance. More specifically, the air sample is expected to have very low temperature and a very high flow rate due to the external ambient temperature and the high speed of the drone (superior to 90 km/h), respectively. The high flow rate can decrease the performance in the case of all sensors, whereas the exterior temperature influence is expected only for the chemical sensors (VOCs, NO_2, O_3). To ensure the desired continuous air flow rate during sampling, a microfluidic MEMS-based chip will be developed and used to create the necessary flow restriction inside our system (Figure 2c). On this chip, the air will be firstly divided in 3 different channels, a central narrower and less deep one and two wider and deeper. Thus, a restricted air flow will be achieved in the central channel, while the rest of the air sample will be exhausted by the two wider and deeper channels. Later on, the central channel is also divided at two, providing two different outlets. Form one outlet the sample will be directed towards the tunnel with T, RH and P sensors (Figure 2a and 2b). From the other outlet the sample will move towards the tunnel of chemical sensors but prior it will pass near the drone's engine, to be heated up, thus enabling the protection of the chemical sensors from low temperatures. Temperature and pressure measured close to the chemical sensors will be used to correct data based on a previous laboratory calibration.



(a)



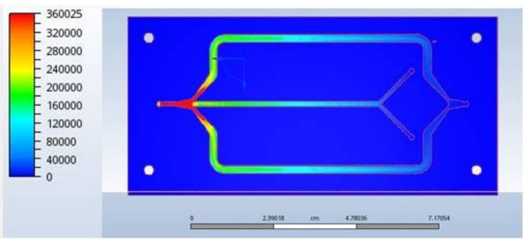
(b)



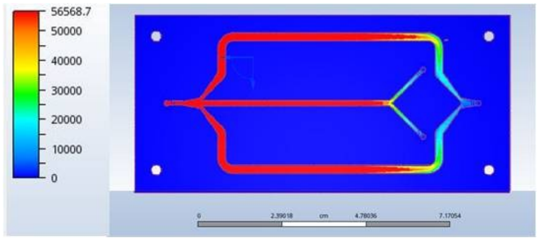
(c)

Figure 2 - Schematic representation of the microfluidic system for sampling during flight (a) and stationary sampling (b); Proposed design of the microfluidic chip for flow restriction inside the system (c)

The dimensions of the microchannels of the microfluidic chip were determined and validated by flow simulations (Autodesk CFD). The simulations were made with a 3D printing polymer as the chip material and for a total inlet of 1 L/min at 0 and 20 °C. Preliminary results indicated that with a central microchannel of 1.00 x 1.00 mm^2 and two external microchannels of 2.25 x 1.00 mm^2 a flow restriction can be achieved so that finally in the central microchannel we have only 22% of the initial flow rate (Figure 3a). This flow restriction is satisfactory for our system since it can protect the sensors while at the same time it enables reasonably quick air renewal. As demonstrated in Figure 3b this flow rate is afterwards equally divided in the two tunnels.



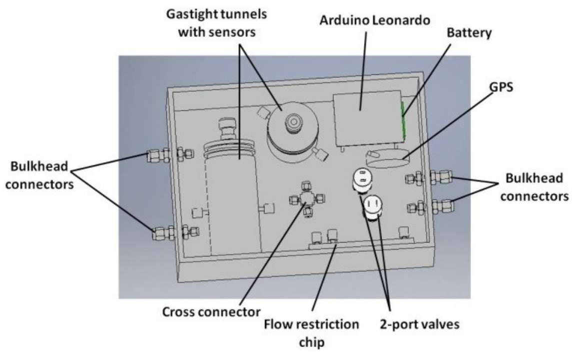
(a)



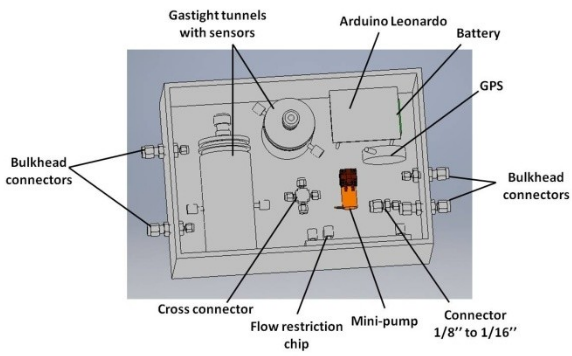
(b)

Figure 3 - Flow simulations on the microfluidic chip: Flow division between external and central microchannel (a); flow division between the two tunnels (b). The scale represents the velocity amplitude.

Necessary electronics and a battery will be integrated to render the system functional and autonomous. The total monitoring system will not exceed 1.5 kg in weight and will occupy a place of 250 x 170 x 105 mm^3 (Figure 4). Furthermore, the energy consumption is expected to not exceed 10 W.



(a)

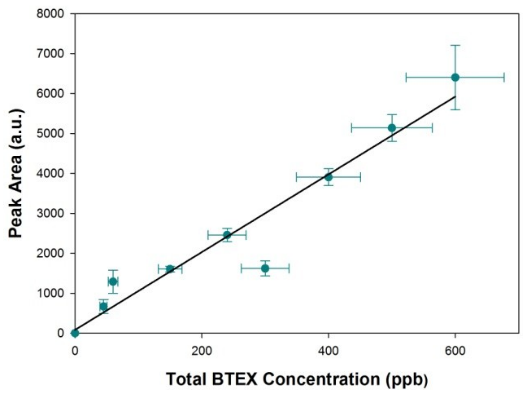


(b)

Figure 4 - 3D design of the total monitoring system for sampling during flight (a) and stationnary sampling (b).

3 SENSORS' EVALUATION

The two metal oxide VOCs sensors (MiCS-5524 and MiCS-VZ-89TE, SGX Sensortech) were tested with a continuous flow of synthetic air, as indicated for their best function. Injections of BTEX (Benzene, Toluene, Ethylbenzene and Xylenes) - a family of VOCs - were made at a volume of 200 µL. BTEX concentration varied in concentrations between 30 and 600 ppb and 3 injections were repeated for each concentration. Figure 5 presents the calibration curves of the VOCs sensors, corresponding to the mean injection peak area as function of BTEX concentration. For both sensors the injection peak area increases perfectly linearly with gaseous BTEX concentration. Detection limits were calculated considering a usual signal/noise ratio equal to 3. For MiCS-5524 the detection limit was calculated to be 44 ppb, whereas for MiCS-VZ-89TE the detection was calculated to be 31 ppb.



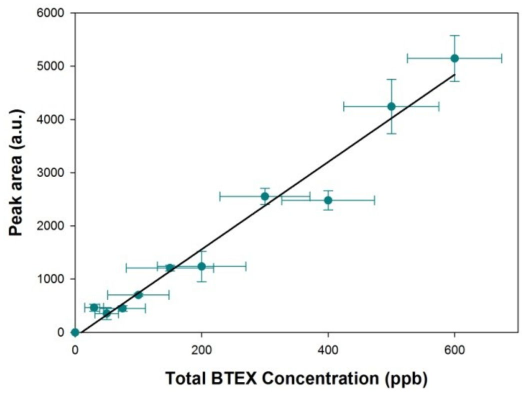


Figure 5 - Calibration curves of VOCs sensors with BTEX. Top: MiCS-5524 (SGX Sensortech); Bottom: MiCS-VZ-89TE (SGX Sensortech). The vertical errors correspond to the standard deviation of peak areas calculated for the three injections. The horizontal errors refer to the uncertainty on the BTEX concentrations, taking into consideration the initial uncertainty of the BTEX cylinder and the precision of flow controllers used for dilution purposes.

4 CONCLUSIONS AND PERSPECTIVES

In this work, we report the conception and development of a sensors-based microfluidic monitoring system for outdoor air quality, meeting all the specifications for integration inside an endurance drone. This new approach presents new possibilities regarding measurements of major outdoor air pollutants and pollution mapping in urban and rural environments.

ACKNOWLEDGEMENTS

This project was supported through the European project ELCOD. The ELCOD project was implemented as a part of the INTERREG V Oberrhein/Rhin Supérieur program and was supported by the European Regional Development Fund (ERDF) and the co-financed project partners Region Grand Est in France and the countries of Baden-Württemberg and Rhineland-Palatinate.

REFERENCES

[1] 'WHO | Air quality guidelines - global update 2005', *WHO*. [Online]. Available: http://www.who.int/phe/health_topics/outdoorair/outdoorair_aqg/en/. [Accessed: 27-Apr-2018].

[2] D. Jacob, *Introduction to Atmospheric Chemistry*. Princeton University Press, 1999.

[3] *Décret n° 2011-1727 du 2 décembre 2011 relatif aux valeurs-guides pour l'air intérieur pour le formaldéhyde et le benzène*. 2011.

Automatic Combination of Line and Point Descriptors for Thermal Aerial Image Mosaicing

L. González-Guzmán*[1], J. Martinez-Carranza[1,2],
[1]Instituto Nacional de Astrofísica, Óptica y Electrónica, Sta. María Tonantzintla, CP 72480, Puebla
[2]University of Bristol, Bristol, UK, BS8 1UB

Abstract

Thermal aerial image mosaicing is a challenging problem, but it becomes more challenging in low-textured images. Thermal images are a particular case where the image may exhibit large portions of the image with low texture. State of the art methodologies using point-based descriptors often obtain poor results when facing the mosaicing task. Motivated by the latter, we propose a novel approach for thermal aerial image mosaicing that automatically combines line-based and point-based descriptors. Line descriptors are used to obtain a fast and robust estimation of the image transformation model, while point-based descriptors are used to support regions without enough line segments information. The proposed approach was evaluated using thermal aerial image sequences captured at 70 meters and comparing results with three of state of the art algorithms for point-based descriptors.

1 Introduction

Image mosaicing is an important process in Computer Vision and Remote Sensing, it has various applications such as: creation of maps, augmented reality, 3D reconstruction, autonomous vehicle navigation, and tracking. Image mosaicing is an active research area in which it seeks to accelerate its process without losing robustness. However, most of the works reported in the literature have focused on the use of visible spectrum images. Consequently, other bands of the electromagnetic spectrum have been little explored. It is important to mention that a visible spectrum image compared with an infrared spectrum image is very different. While visible spectrum images represent information with light, color and texture; infrared images represent information from the temperature of objects, forming blobs or regions with similar intensities in the image, consequently these images have little texture and absence of colors in the scene [1]. Methods of image mosaicing based on points that use thermal images, generally do not obtain good results due to the lack of texture in this kind of images.

*Department of Computer Science at INAOE. Email addresses: {luis-gonzman, carranza}@inaoep.mx

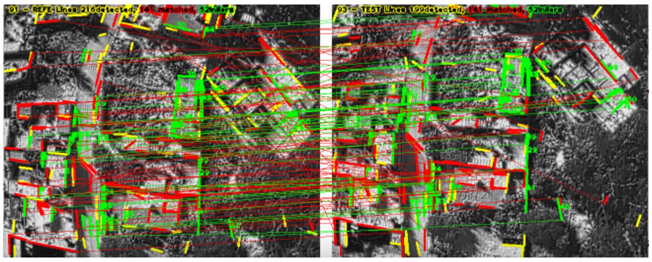


Figure 1: Line segments correspondences between a reference image (left) and a test image (right). Yellow lines represent lines without correspondence. Red lines show wrong matches. Green lines represent inlier matches (good matches). https://youtu.be/TdJ2Xj8nSxo

This research proposes a robust and efficient alternative for the creation of thermal aerial image mosaics using line primitives. Using line segments and their descriptors, it is possible to obtain a reliable description in low texture images, since each line segment is represented by a larger region when it is compared to the region representing by a point. In addition to the extension of a line segment, which represents more information in the discrimination of features, the geometric information, such as angles and intersection points, are used as criteria to obtain an image transformation model namely, a homography. Figure 1 shows a pair of images from which lines were extracted and correspondences were searched to determine a transformation model between two images.

On the other hand, we present an automatic and fast approach based primarily on quadtrees to decide what keypoints should be used to support the line segments. The main idea is to use only those points lying on image areas where no line intersections are located, seeking to make the most of all the available information in the image, this is, extracting points only in certain missing features regions and not in the whole image as would normally be done in a blind strategy, thus reducing the execution time and providing much less points, but quaility points, for the estimation of the transformation model, see Figure 2.

Experiments carried out in this research concentrate on the comparison of point-based techniques with respect to line-based techniques using the proposed approach for thermal image mosaicing. The results obtained show that the proposed methodology is an automatic and efficient alternative

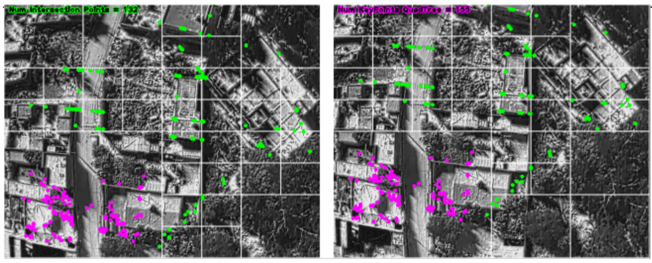


Figure 2: Point correspondences: Green points represent intersection points from inlier line segment correspondences (green lines from Figure 1) and pink points show quadtree-keypoint matches in areas where intersection points were not found.

for thermal aerial image mosaicing. In addition, experiments carried out indicate that our approach is faster than the techniques based on primitive points such as SIFT [2], SURF [3] and ORB [4]. In the case of ORB algorithm that has been commonly used in real-time applications, the proposed approach obtains better results in the final mosaic because due to the adequate selection of features, either line only, points only or both, to be used to generate the homography required to carry out the stitching task.

2 Related work

Image mosaicing in the visible spectrum is a well-studied area where there are numerous works that try to obtain a balance by reducing the execution time and conserving robustness. However, in the literature few works address the problem of image mosaicing in low texture visible or IR images. Some related works presented below that make use of line segments or keypoints in visible or infrared spectrum images.

2.1 Line segments for image mosaicing in visible spectrum images

J. Zhu and M. Ren [5] propose a feature matching method based on SIFT to extract points and form targeted line segments. This method uses the Harris corner detector to extract keypoints in order to build graphs directed from the extracted points. Subsequently, it describes directed line segments using SIFT algorithm and compares them with the line segments determined in another image to achieve the approximate correspondence of points. Finally, the matching points are adjusted and the erroneous pairs are eliminated through the RANSAC method to achieve the mosaic of images. The line segments description of the proposed method continues to preserve the robustness of SIFT algorithm to transformations such as image rotation, distortion and scaling, as well as illumination. However, this method proposed by J. Zhu and M. Ren, has some disadvantages in low texture images such as thermal images, because the techniques they use to extract features (Harris and SIFT) are mainly based on the extraction of information of corners and textures. A similar approach is carried out by Z. Yang et al. [6], where the authors propose to use keypoints detected by SURF to construct directed line segments, the approach proposed by the authors improves the execution time. However, the techniques they use to extract keypoints and form lines depend on regions rich in texture, therefore, their proposal is not robust in images of low texture or thermal images.

2.2 Points for image mosaicing in thermal images

The approach proposed by P. Shah [7] uses thermal images to generate mosaics by implementing a lens aberration correction and a brightness correction method. The developed method uses SIFT for the extraction and description of keypoints due to its robustness features in most transformations. For the estimation of the homography, the author used RANSAC algorithm to eliminate outliers. However, its method is not very accurate in images with little overlap because SIFT does not obtain a sufficiently good set of key points in the thermal images.

On the other hand, Y. Wang et al. [8] performs the creation of mosaics from frames of video captured by a thermal camera. The detection of low level features was performed using SIFT algorithm. The authors comment that although SIFT is one of the most robust methods in the literature, it has problems with the small number of common points between pairs of thermal images and traditional methods such as "Least Square" induce the erroneous registration of correspondences . To deal with this problem Y. Wang et al. propose an easier to implement and highly robust method called "Random M-least square" as a replacement for bundle adjustment. The proposed method is partially related to least squares and also maximizes the number of points of adapted features that fit the transformation model within an accepted registration error. The use of their proposed algorithm increases the correspondence of the points detected by SIFT but does not attack the feature detection problem, so its algorithm is limited to the set of points that the feature extractor detects, this makes that its method work well when SIFT finds a broad set of key points.

2.3 Line segments for image mosaicing in thermal images

In the literature there are no related works where line segments are used directly for the creation of mosaics in thermal images. However, there is a work published by L. Wei et al. [1] that uses line primitives to only estimate the affine matrix as a transformation model between the images. This approach proposes a method of eliminating matches of erroneous line segments, it consists of two steps: The calculation of the affine transformation matrix by intersection points of three randomly selected lines using a modification of RANSAC algorithm, and finding the best transformation by a match score ordering. Once the authors calculate the transformation matrix, they evaluate each line segment by transforming it and measuring the error of the transformation with the center point of the line and a tolerance angle, if this error is less than a threshold, the line segment it will be considered as an inlier, otherwise as an outlier. This proposed method has certain drawbacks in not considering perspective transformations between the images and another important aspect is that the central distance of a line is not a reliable metric because the size the line segment may vary between images.

Finally, this research uses line segments to estimate a transformation model and quadtree-keypoints are used as support when no line was detected or when there is not enough line information. Details of our proposal are shown in Section 3.

3 Methodology

This research uses intersection points from line segments to estimate a fast and robust transformation model. Subsequently, the intersection points of the lines labeled as inliers by our RANSAC algorithm are used to generate qudtrees of sub-regions from the image, with the aim of locating regions without sufficient intersection points (green regions, see Figure 4(b)) according to an established threshold. If the percentage of green regions is greater than 60%, keypoints are extracted to obtain a transformation model based on points (see Figure 4(c)) and a comparison is made using the keypoints detected between the line model, points and the combination of both, always looking for the best model that obtains greater number of inlier keypoints. On the other hand, if the number of inliers is relatively close, the model with smaller standard deviation is chosen. Figure 3 shows a diagram of the proposed approach, each stage is described below.

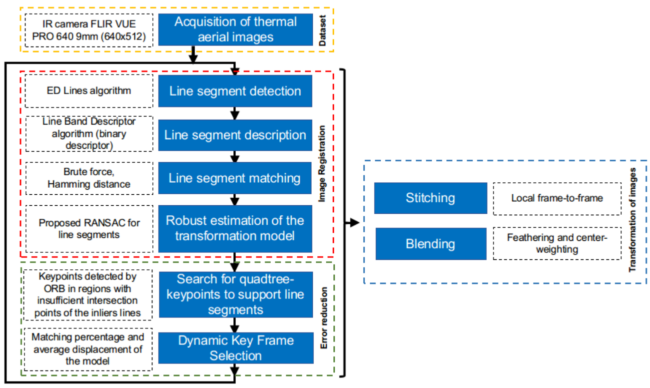


Figure 3: Proposed approach

3.1 System overview

The algorithms developed in this research were implemented in C++ using GNU Compiler Collection (GCC) 5.4.0 and using Open Source Computer Vision Library (OpenCV) 3.3.0 on a personal computer with an Intel Core i7-4720HQ (2.60GHz) processor and 16GB of RAM memory.

3.2 Dataset

Sequences of thermal aerial images were captured using a FLIR VUE PRO 640 infrared camera mounted on a UAV, the images have a spatial resolution of 640x512 pixels. Some of the aerial thermal sequences acquired in this research can be downloaded in the following link: Thermal Aerial INAOE Dataset.

3.3 Line segment detection, description and matching

The detection of line segments was performed by ED-Lines [9] algorithm, which obtains segments of a pixel of thickness and it is considered one of the fastest algorithms to extract this kind of feature. On the other hand, the description of line segments was performed by the Line Band Descriptor [10] algorithm, this descriptor uses bands (sub-regions) near the line segment to extract information by means of local and global convolutional masks, finally it generates a binary vector of mean and standard deviation with the information of the bands. Once line segments were described and detected, brute force and Hamming distance were used to determine correspondences between the pair of images.

3.4 Elimination of erroneous correspondences and robust model estimation

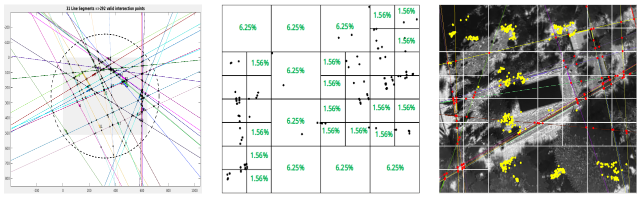
These stages were carried out by a proposed adaptation of RANSAC [11] algorithm for line segments. Similar to the work of L. Wei et al. [1], we use intersection points to estimate the transformation model and eliminate erroneous correspondences. Using minimum distance between line segments and using two transformation error thresholds (distance and angle), it is possible to determine inlier line segments and estimate a robust model that uses geometric information of the structures present in the image. Once the best model was found, our algorithm recalculates the transformation model using all the intersection points of the inlier lines as long as their transformation error is less than or equal to 3 pixels and they are within an established range. Figure 4(a) shows an example of valid intersection points used to recalculate the model.

3.5 Search of support keypoints

In this stage quadtrees were used to identify regions with insufficient intersection points (less than 4 points per quad), see Figure 4(b). If the percentage of quads with insufficiency of intersection points is greater than 60% of the complete image then keypoints of support will be extracted only in these regions and a model based on keypoints will be estimated, otherwise the line model will be used only. The main objective of this stage is to extract points distributed throughout the image and preserve the best model obtained from line segments, keypoints or a mixture between lines and points (the average of line-based and point-based models), which best describes the relationship of the images, see Figure 4(c).

3.6 Transformation of the images

Traditionally in the stitching process, the approaches proposed in the literature consider all the images of the input frames set even if the camera remains immobile or if the change between frames is minimal. It is important to consider



(a) (b) (c)

Figure 4: Search for quadtree-keypoints to support line segments. (a) Valid intersection points. (b) Quadtree generated, green quads represent regions where keypoints will be extracted. (c) Intersection points (red) and quadtree-keypoints (yellow).

that the execution time is high and that errors in the estimation of the homography are propagated for the following ones that fit the canvas because each model depends on the previous ones to form the transition of the sequence of images in the final mosaic. For this reason, this stage was performed frame-to-frame using Dynamic Key Frame Selection (DKFS) proposed by J. Li et al [12] to reduce execution time and accumulated error in the mosaic, see Figure 5. In our DKFS implementation, we consider a relevant frame if its match percentage is less than 40% or if the average displacement of the model is greater than 20% of the image's size. Furthermore, the feathering and centered-weighting techniques were used to fade borders between regions with coincidence in the final mosaic.

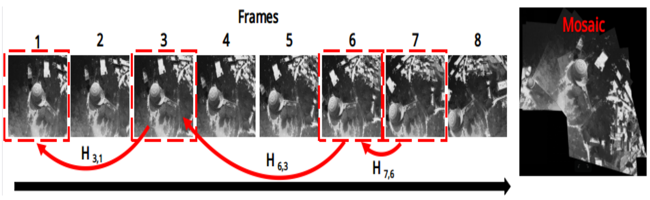


Figure 5: Dynamic Key Frame Selection used to select relevant frames according to sequence transition.

4 Experimental results and analysis

Experiments were performed on a sequence of thermal aerial images composed of 335 frames, this sequence was captured by an UAV on a rectangular route of approximately 660 meters, see Figure 7(a).

In order to evaluate the proposed approach, the comparison of performance, execution time and quality of the final mosaic were made with three of the main algorithms for keypoint detection of state-of-the-art. Some tests were performed using keypoints detected by SURF, SIFT, and ORB algorithms in order to create mosaics of thermal aerial images. On the other hand, the same test was performed using only line segments, and using the proposed methodology combining line segments in conjunction with quadtree-keypoints detected by the three keypoint detection algorithms.

4.1 *Inliers error evaluation*

Table 1 shows the performance obtained in each test, evaluating the average number of features detected, matches, inliers determined by our RANSAC algorithm, average distance error (minimum distance for line segments), average angle error (always 0 if points were evaluated), and finally, the number of frames considered in the mosaic by our DKFS algorithm. In the evaluation, 2000 features were established as limit in the detection of each technique. On the other hand, a feature inlier was defined if it satisfies the following conditions: It has 3 pixels of error or less (points and lines) and it has 3 degrees of error or less (lines only).

Table 1: Performance comparison.

Execution	Average Number			Average Error				Number of frames used in the mosaic
				Distance		Angle		
	Features	Matches	Inliers	Mean	SD	Mean	SD	
SURF	1994	1325	711	1.466	0.727	0.000	0.000	115 / 335
SIFT	1994	1324	670	1.390	**0.724**	0.000	0.000	120 / 335
ORB	1994	1316	726	1.580	0.737	0.000	0.000	93 / 335
LBD	175	124	61	**0.970**	0.809	0.796	0.553	103 / 335
LBD-QT-SURF	1102	668	278	1.305	**0.751**	**0.195**	**0.140**	152 / 335
LBD-QT-SIFT	1011	636	246	1.228	0.758	0.299	0.212	142 / 335
LBD-QT-ORB	499	304	135	**1.215**	0.780	0.421	0.292	116 / 335

The results presented in Table 1 show that the test performed only with line segments (LBD) obtains the lowest average error in distance but the worst standard deviation of distance, also, the average error of angle obtained is the worst of all the tests. On the other hand, the approaches based on the use of line segments and quadtrees-keypoints (LBD-QT-SURF, LBD-QT-SIFT and LBD-QT-ORB) reduce the error of angles compared with the approach of only line segments. Likewise, the test performed with line segments and quadtrees-keypoints detected by ORB (LBD-QT-ORB) obtains the best performance in terms of average distance error, while the lowest average angle error is obtained by the test performed with line segments and quadtrees-keypoints detected by SURF (LBD-QT-SURF). Besides, Table 2 shows the list of models used in the 334 iterations performed by each test.

Table 2: Comparison of the type of model selected in each iteration.

Execution	Type of model selected (Number of iterations)		
	Only Points	Only Lines	Mixed Model
SURF	334	0	0
SIFT	334	0	0
ORB	334	0	0
LBD	0	334	0
LBD-QT-SURF	246	86	2
LBD-QT-SIFT	198	132	4
LBD-QT-ORB	**149**	**180**	**5**

The results show that the test carried out with line segments and quadtrees-keypoints detected by ORB shows a balanced pattern between the use of line and point models, this pattern is the expected one due to the fact that half of the scene lines are abundant (for buildings, roads, etc.), while in the other part of the scene there are very few lines for the vegetation present, see Figure 7(a).

4.2 *Run time evaluation*

Table 3 shows the average execution times of each stage of image mosaicing according to the previous tests. The results show that the test performed with only line segments is the one with the shortest execution time. Tests performed where points are used without quadtrees (SURF, SIFT and ORB), consume too much execution time for the creation of mosaics. On the other hand, line segments in conjunction with quadtrees-keypoints (LBD-QT-SURF, LBD-QT-SIFT and LBD-QT-ORB) reduce the execution time in all the tests due to the fact that quadtrees locate sub-regions where it is necessary to extract points to support the lines, instead of extracting them throughout the image as originally done in the literature. Figure 6 shows a comparative graph of the average run time per frame of each test.

Table 3: Average run time per frame in each stage of image mosaicing.

Execution	Average Run Time (milliseconds)							
	Reading	Detection and Description	Matching	RANSAC	Quadtrees	Model Adjustment	Stitching and Blending	Total Average Run Time
SURF	4.474	87.199	39.972	349.077	0.000	0.004	0.081	480.806
SIFT	1.203	197.101	77.962	357.441	0.000	0.004	0.052	633.764
ORB	1.045	**21.167**	17.223	369.448	0.000	0.003	0.064	408.949
LBD	0.989	27.727	**0.325**	**147.741**	0.000	0.006	0.085	**176.873**
LBD-QT-SURF	0.974	27.738	0.337	151.710	255.667	0.009	0.053	436.488
LBD-QT-SIFT	0.965	27.985	0.342	149.205	406.651	0.008	0.055	585.212
LBD-QT-ORB	0.979	27.926	0.338	148.598	**102.552**	0.007	0.071	**280.472**

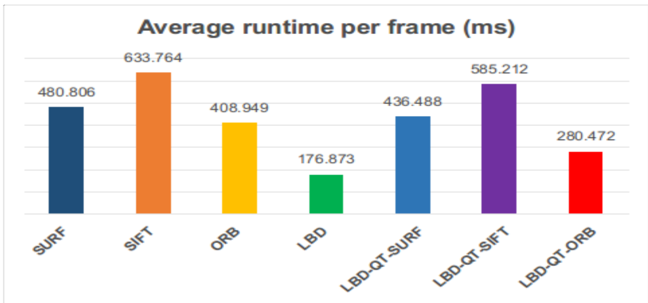


Figure 6: Comparison of average run time per frame in milliseconds.

4.3 *Qualitative evaluation of the mosaics*

Figure 7 presents the mosaics generated by each test performed. The results show that the use of line segments is not enough for scenarios with a lot of vegetation because the accumulated error affected the aligning of the initial and final parts of the mosaic, see Figure 7(b). On the other hand, the tests where only points are used, like the lines, they present error in certain regions of the mosaic generated by the great ambiguity of intensity in regions with vegetation. Tests done with only keypoints (SURF, SIFT, and ORB) show an approximate reconstruction to the real world scene, however, the runtimes of these tests are very high, see Table 3.

Finally, tests performed with line segments in conjunction with quadtrees-keypoints (LBD-QT-SURF, LBD-QT-SIFT, and LBD-QT-ORB) also show error in some regions of the mosaic. In the case of the test performed with line segments and quadtrees-keypoints detected by ORB (LBD-QT-ORB), see Figure 7(h), it is possible to notice that there are effects of ghosting but this test retains better the shape of the reconstructed region, this is because ORB algorithm determines keypoints near the contours of the objects present in the scene while intersection points of the line segments provide information in the outlines and outside of them, see Figure 2.

5 Conclusions

Image mosaicing in thermal images is an area with great research opportunities, where the majority of state of the art algorithms focus on the use of point-based descriptors. These descriptors are successful in color images, where texture and color patterns are exhibited. In contrast, thermal images lack this information and therefore, we have presented a robust and efficient alternative to solve the problem of image mosaicing in thermal aerial images using line segments in conjunction with a search strategy for support points.

Thus, we have described a novel approach for the generation of thermal aerial image mosaics. Our proposed framework decides in each iteration whether to use a model based on points, a model based on lines or a mixture of both, in order to take advantage of information coming from the structures present in the scene, thus obtaining a model that best describes the relationship between images, without requiring to extract several points in the whole image, hence reducing execution time.

According to our experiments, our proposed approach obtains comparable results with respect to state of the art techniques based on points. We also obtain less execution time and less transformation error.

As a future work, we will push for an optimize version of our approach in order to achieve real-time thermal image mosaicing, also including transferring our methodology to visible images.

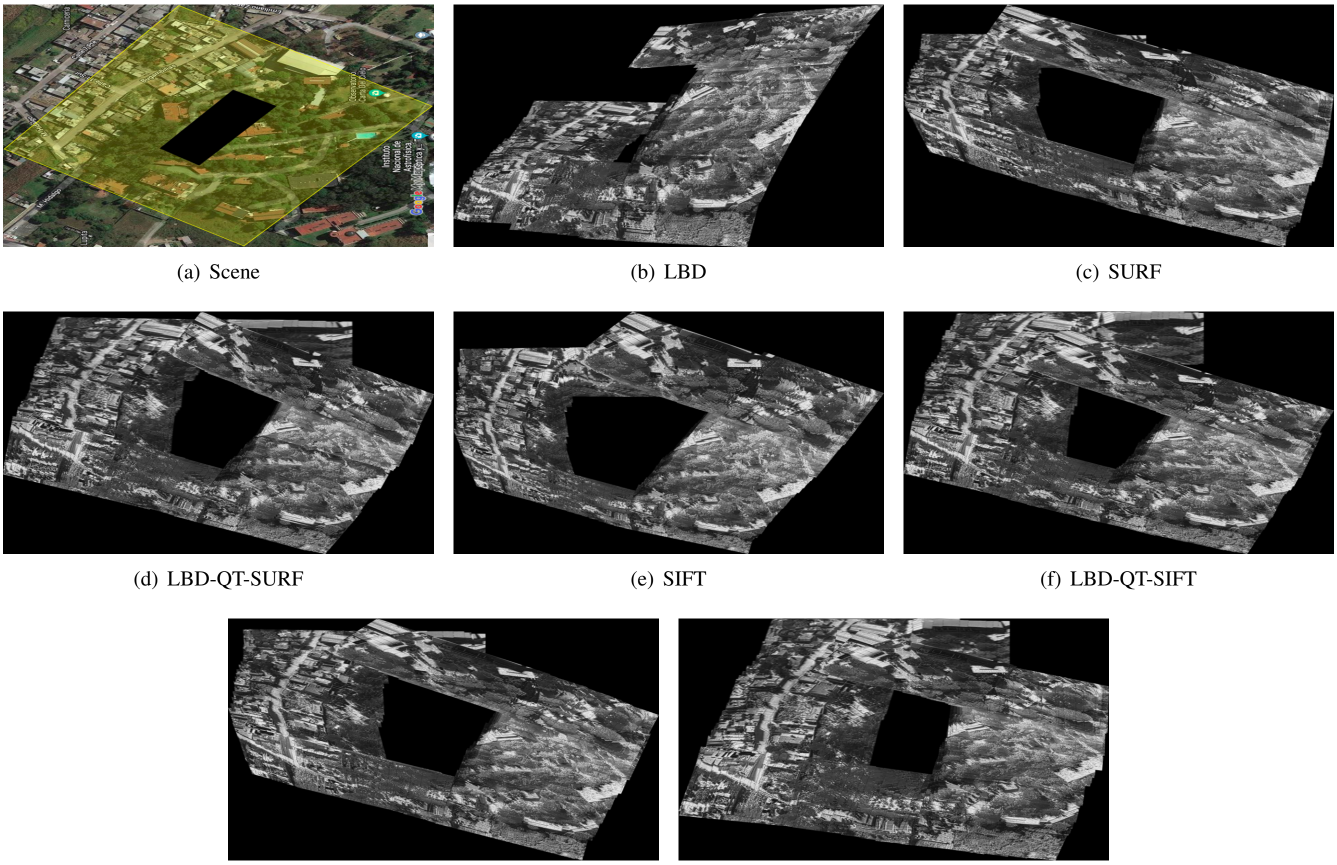
6 Acknowledgements

This research has been supported by CONACYT grant 608901 and CONACYT-INEGI project 268528.

References

[1] Lijun Wei, Yong Li, Hang Yu, Liangpeng Xu, and Chunxiao Fan. An improved RANSAC algorithm for line matching on multispectral images. *Proc.SPIE*, 10225:10225 – 10225 – 5, 2017.

[2] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.



(a) Scene (b) LBD (c) SURF

(d) LBD-QT-SURF (e) SIFT (f) LBD-QT-SIFT

(g) ORB (h) LBD-QT-ORB

Figure 7: Comparison of mosaics generated using line segments, keypoints and line segments in conjunction with QuadTree-keypoints.

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 2008.

[4] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[5] Jun Zhu and Mingwu Ren. Image mosaic method based on SIFT features of line segment. *Computational and Mathematical Methods in Medicine*, 2014:11, 2014.

[6] Zhanlong Yang, Dinggang Shen, and Pew Thian Yap. Image mosaicking using SURF features of line segments. *PLoS ONE*, 12(3):1–15, 2017.

[7] Preeyanka Shah. Image Processing Aerial Thermal Images to Determine Water Stress on Crops. Technical report, Stanford University, Stanford, CA, 2013.

[8] Yi Wang, Aldo Camargo, Ronald Fevig, Florent Martel, and Richard R. Schultz. Image mosaicking from uncooled thermal IR video captured by a small UAV. *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 161–164, 2008.

[9] Cuneyt Akinlar and Cihan Topal. EDLines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 32(13):1633–1642, 2011.

[10] Lilian Zhang and Reinhard Koch. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. *Journal of Visual Communication and Image Representation*, 24(7):794–805, 2013.

[11] Martin A. Fischler and Robert 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.

[12] Jing Li, Tao Yang, Jingyi Yu, Zhaoyang Lu, Ping Lu, Xia Jia, and Wenjie Chen. Fast aerial video stitching. *International Journal of Advanced Robotic Systems*, 11, 2014.

Vision-based Autonomous Navigation for Wind Turbine Inspection using an Unmanned Aerial Vehicle

R. Parlange*[1], J. Martinez-Carranza[1,2], L.E. Sucar[1], and B. Ren†[3]

[1]Instituto Nacional de Astrofísica, Óptica y Electrónica , Luis Enrique Erro 1, Sta. María Tonantzintla, CP 72840, Puebla
[2]University of Bristol, Bristol, UK, BS8 1UB
[3]Texas Tech University, 2500 Broadway, Lubbock, TX 79409

ABSTRACT

Wind turbines require periodic inspection and maintenance to ensure good performance and a prolonged lifetime. Traditionally, inspection involves the risk of a person falling while climbing down from the platform. Trying to eliminate this risk, Unmanned Aerial Vehicles (UAVs) have been controlled by operators to inspect the structure while taking pictures and video. In contrast, we propose an autonomous UAV system that is able to locate itself and build a map of its environment using visual SLAM. Perception of static rotor blades is based on a single observation, where the Hough transform for lines is used to detect the position of the hub and the angle of the blades, allowing the path planner to make a backwards projection from the 2D image plane to the 3D scene, establishing a set of waypoints to inspect the surface from a safe distance. Experiments were carried out in a simulated environment and a real setting.

1 INTRODUCTION

Wind turbine inspection has been traditionally performed through simple visual inspection from the ground with a telephoto camera lens or by a person who climbs down from the platform. The former method is time-consuming and the latter puts a human life at risk; while both are restricted by the mobility and field of view of the operator.

Trying to overcome these challenges, a recent approach using Unmanned Aerial Vehicles (UAVs) is being adopted in the industry. In contrast to traditional methods, an UAV offers an increased mobility and a close-up view of the surface of the blades. However, this task requires expert pilots and causes them to experience fatigue quickly. Alternatively, autonomous UAVs are not subject to human tiredness and can follow trajectories in a repeatable manner.

Figure 1: Left column: simulated inspection using Gazebo, with the bottom image showing the view of the frontal camera; right column: likewise, inspection carried out on an emulated turbine. https://youtu.be/XZZm345rCRY

Autonomous inspection of wind turbines poses a series of challenges. The first one is the localization of the UAV within its environment. This was approached with a computer vision technique named Simultaneous Localization and Mapping (SLAM), to estimate the pose of the UAV, by extracting features from the monocular input of its camera and anchoring them into a map it builds of its surroundings.

To perform the inspection, the UAV must perceive the wind turbine and determine its position relative to the frame of the UAV. Based on an arbitrary takeoff position, the UAV must detect the hub of the wind turbine and the angle of its blades using line features. Once detected, the path planner establishes an inspection trajectory for the UAV to follow while it takes pictures and video of the surface from a safe distance. This method aims at the automatic acquisition of high quality image data for further evaluation with human expertise.

*Department of Computer Science at Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE).
Email addresses: {parlange, carranza, esucar}@inaoep.mx
†Department of Mechanical Engineering at Texas Tech University.
Email address: beibei.ren@ttu.edu

2 Theoretical framework

2.1 Localization

SLAM with a monocular camera can only build a map up to a scale factor. Autonomous navigation requires a metric scale for path planning, that takes into account the dimensions of the object of interest and the scene. To address this problem, the RGB-D version of ORB-SLAM2 [1] was used, where the RGB frames are coupled with a synthetic depth map [2]. Figure 2 shows the generation of the metric map using the above mentioned procedure. The detection, planning and autonomous navigation modules are described below.

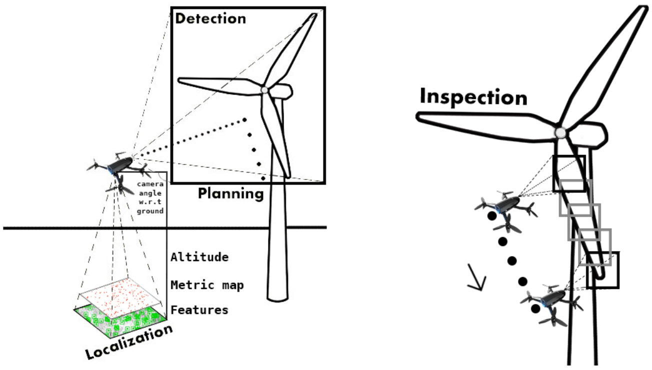


Figure 2: Inspection diagram with three main modules: localization, detection, planning and autonomous inspection.

2.2 Detection

Computer vision tasks such as object recognition require an efficient representation, reducing the amount of data in the image and preserving its visual characteristics and structural information. In order to detect straight lines with the Hough transform, edges must be detected beforehand. Edge detection reduces the amount of data to be processed while conserving the outline of the wind turbine. However, some scenes require further processing to remove image noise. The HSV color model was used to segment the image by separating hue, saturation and value channels.

A Gaussian filter was used to blur and remove noise from the image. The Canny edge detector [3] calculates the directional derivatives with the Sobel operator, which uses two 3x3 kernels, one for horizontal and one for vertical differentiation. These kernels are convoluted with the original image, to obtain the magnitude and direction of the gradient. The output is a binary image with the edges, suitable for straight line detection.

The Hough transform, uses an angle-radius parametrization [4], instead of the original slope-intercept parameters [5] to detect the most prominent straight lines [6] from its camera input, as shown in Figure 3. The position of the hub can be found by looking for the intersection of the blade lines.

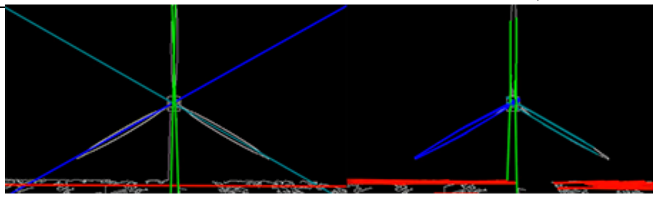


Figure 3: Standard (left) and probabilistic (right) Hough transform for lines, segmented by angle.

2.3 Autonomous navigation

To describe the mathematical relationship between the coordinates of one point in three-dimensional space and its projection onto the image plane, we use the pinhole camera model. With this ideal model, we can also obtain the 3D coordinates from a 2D image if the depth of the image and the intrinsic parameters of the camera are known.

A backward projection is done to recover 3D coordinates from the detection pixel in the 2D image plane, given a known depth [1]. The path planner uses the coordinates from the detection of the hub, along with the angles of the blades to create a flight plan.

Once the path planner has established a set of inspection trajectory points, the UAV must follow these waypoints to approach the blades of the wind turbine and capture photos/video of its surface. In order to accomplish this task, we must find the error between the current pose of the UAV and the target 3D coordinate or setpoint in the control loop.

The UAV approaches the hub of wind turbine using a longitudinal approach. Then it switches to lateral movements to keep the camera oriented perpendicular to the surface of the blades. This process is implemented as a finite state machine with three states: altitude, rotation, and translation.

Altitude. The first state is a proportional (P) controller for the altitude, that minimizes the error between the current z-axis position and the reference, established by the waypoint. Once the error is less than the threshold, the state machine transitions to the rotation state.

Rotation. In this state, the current rotation is compared to a reference rotation matrix that maintains the orientation towards the surface of the blades. A proportional integral (PI) controller is used to reduce the error between the current rotation and the setpoint.

Translation. The last state is a translation controller. The rotation state keeps the orientation of the UAV fixed towards the blade, while a proportional integral (PI) control loop reaches the desired y-axis position.

[1] Depth corresponds to the arbitrary distance from the takeoff position to the hub of the wind turbine.

The controller iterates between these states until the control signals minimize the error on all three axes within a certain threshold. Once this condition is met, the UAV proceeds to follow the next waypoint. Figure 4 shows this navigation controller as part of a state machine of the autonomous inspection procedure described beforehand.

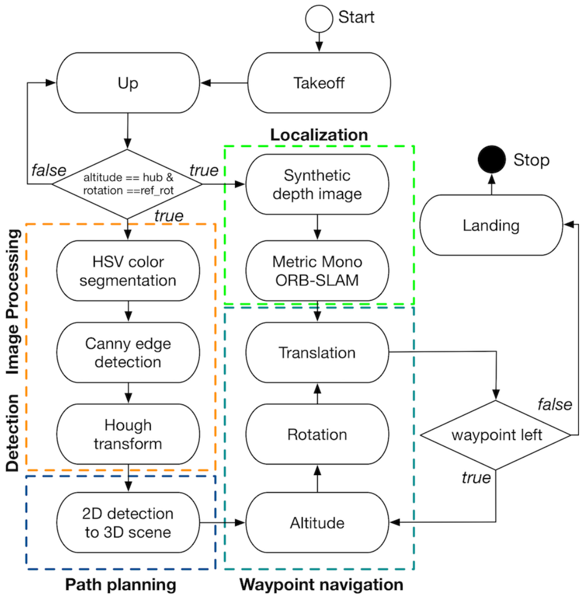


Figure 4: Flowchart of the finite state machine.

3 Experimental Framework

3.1 System Overview

The robotic platform used to carry out the experiments was a Parrot Bebop 2 for the physical trials and a Parrot AR.Drone 2.0 in simulation. Even though the UAVs have different Software Development Kits (SDKs), *bebop_autonomy* and *ardrone_autonomy* share topic names, types and coordinate frame conventions for core piloting tasks. These shared characteristics allow the development of algorithms in a simulated environment before testing the effects of the same program in a real setting. As opposed to the AR.Drone 2.0 simulated UAV, the Bebop platform only has a frontal monocular camera. This camera is used for localization, so an additional GoPro camera was attached to the back of the UAV to record the inspection.

The Kinetic distribution of Robot Operating System (ROS) runs on top of an Ubuntu 16.04 Linux operating system. Both *bebop_autonomy* and *ardrone_autonomy* are available as ROS drivers. In order to test and tune our visual navigation system we use Gazebo, a ROS-based robotics simulator. In particular, we use the *tum_simulator*, a implementation of the Gazebo simulator with a model of the AR.Drone 2.0, developed by Hongrong Huang and Jürgen Sturm from the Computer Vision group at the Technical University of Munich (TUM).

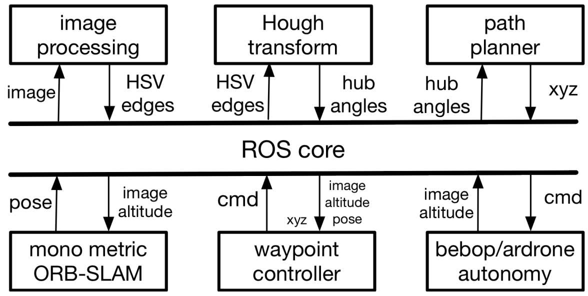


Figure 5: The publisher/subscriber ROS architecture is the same for the simulated environment and real setting scenes.

4 Results

Experiments were carried out in simulation using a model of a scale wind turbine with a height of 10 m to the hub, and in a real setting we used a 3 m tall, emulated scale wind turbine.

4.1 Simulated environment

The waypoint controller follows the inspection trajectory, which is generated by the path planner based on the detection of the hub and the angles of the blades. This control loop uses the continuous localization from the metric SLAM system to estimate the pose of the quadrotor and navigate around the blades of the wind turbine, taking pictures of its surface. Figure 6 shows a 3D plot of the pose estimation from ORB-SLAM against the ground truth from the Gazebo simulator. As can be observed, the inspection trajectories match in all three axes, proving the capacity of the localization system to retrieve an absolute metric scale from its monocular input. The x-axis or depth corresponds to the arbitrary distance to the hub which permits the backwards projection with the pinhole camera model.

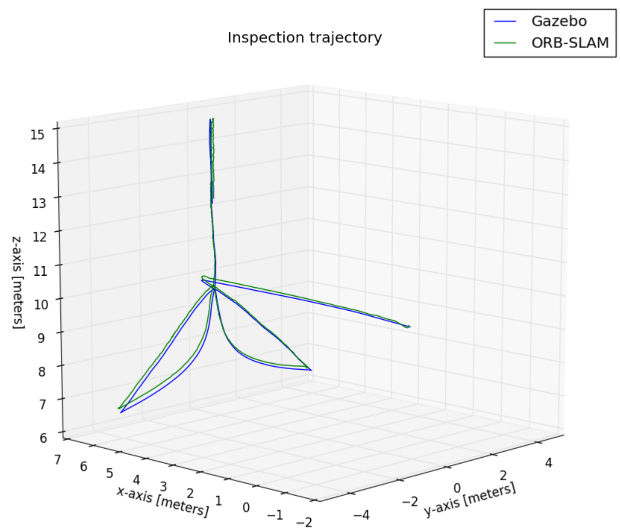


Figure 6: Gazebo ground truth vs. SLAM pose estimation. Root mean squared error, RMSE = 0.344.

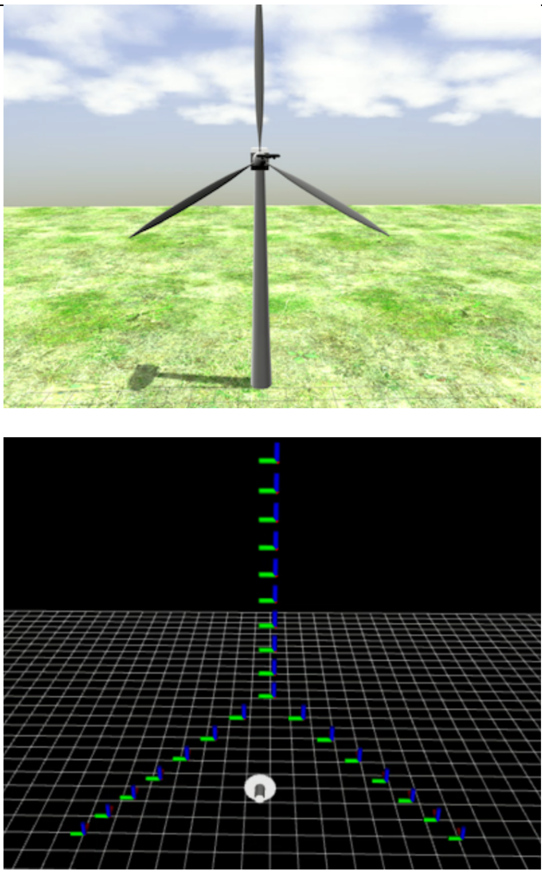


Figure 7: (a) Gazebo simulation. (b) Inspection plan in rviz.

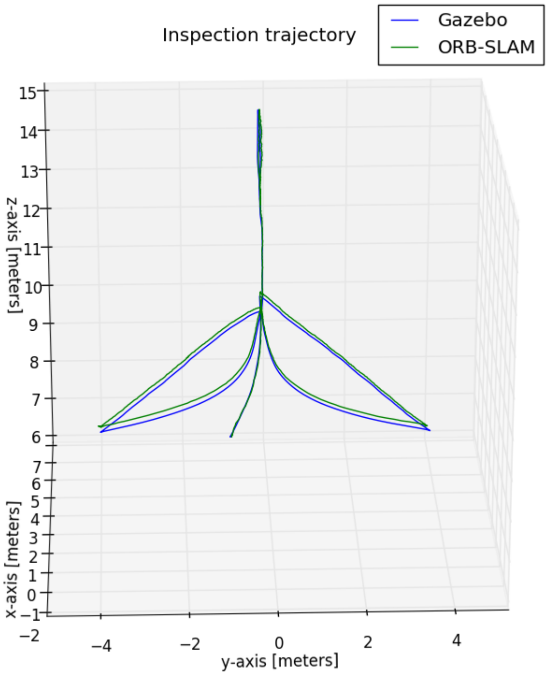
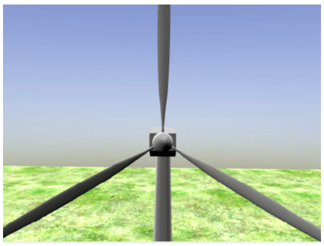
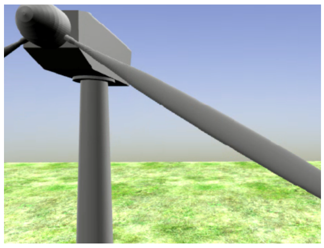


Figure 8: Frontal view of the inspection trajectory.

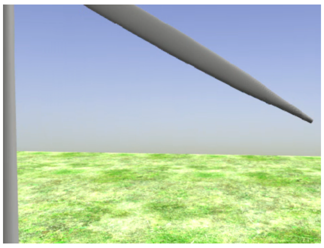
Figure 7(a) shows a Gazebo simulation with the model of the AR.Drone 2.0 inspecting a scale wind turbine with a height of 10 m to the hub or rotor. Figure 7(b) displays the waypoints in the ROS visualization package, rviz. The white arrow depicts the actual pose of the quadrotor estimated by the metric SLAM system. Figure 8 is a frontal view of the trajectory for comparison with the inspection plan above.



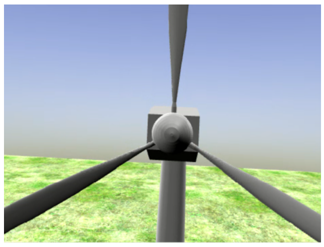
(a) Simulation - Hub.



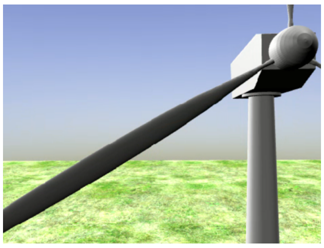
(b) Simulation - Right blade.



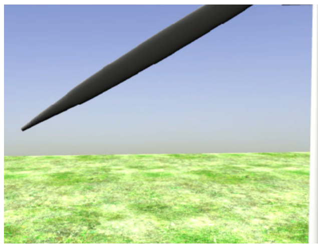
(c) Simulation - Tip of right blade.



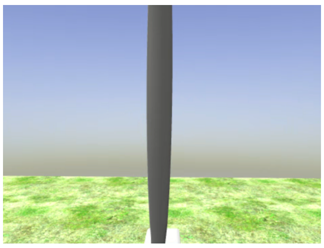
(d) Simulation - Hub.



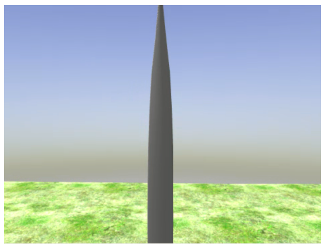
(e) Simulation - Left blade.



(f) Simulation - Tip of left blade.



(g) Simulation - Top blade.



(h) Simulation - Tip of top blade.

Figure 9: Wind turbine inspection in simulation.

The sequence of pictures in Figure 13 shows captures taken at the generated waypoints. The purpose of this image data is to provide useful information for an inspection expert, to detect damage on the surface of the blades and schedule maintenance for the wind turbine, to sustain its performance.

4.2 *Real setting*

Due to illumination changes and background image noise found in the proposed scene, alternative color segmentation and detection methods were used.

The CIE L*a*b color space was used to segment the wind turbine from the background. It expresses color in three numerical values, *L* for lightness, *a* for green-red and *b* for blue-yellow components, with a perceptually uniform distribution with respect to human color vision. In this particular scene, the color model allowed the distinction of the white wind turbine from the glow of the sky and the reflection of the light on the leaves. Thus, removing noise and enhancing the detection of the blade lines.

LSD: A Line Segment Detector [7] had better line detection results as it requires no parameter tuning, making it suitable for detection at different lighting conditions. Figure 10(a) shows the six prominent finite line segments, corresponding to the blade lines, extended to infinite lines. Subsequently, the hub is detected by obtaining the centroid of the intersections of these lines. This coordinate is then used by the path planner, which makes the backwards projection to generate the inspection points along the blade lines, as depicted in Figure 10(b).

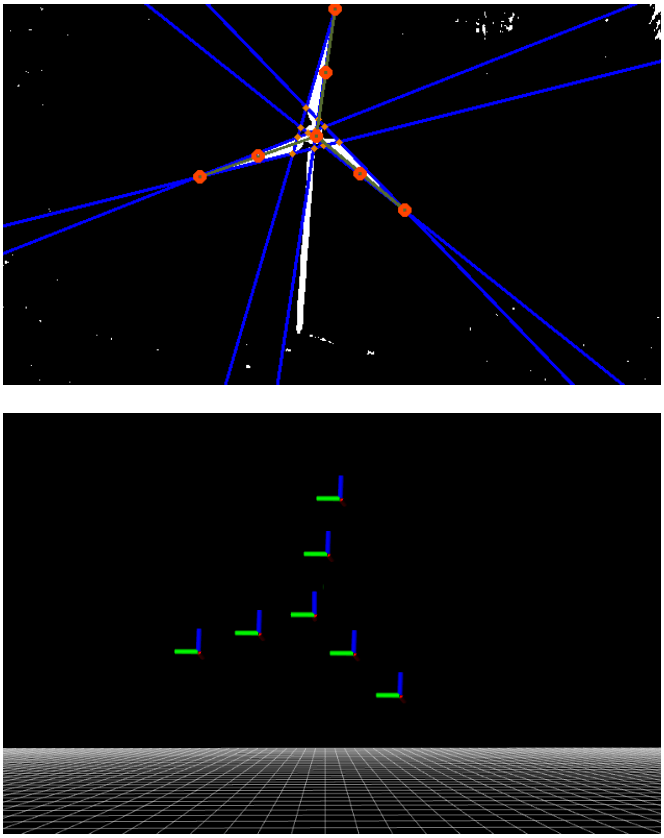


Figure 10: (a) Hub and blade detection over the CIE L*a*b color space. (b) Inspection plan in rviz.

The UAV carried out the autonomous inspection of a 3 meters tall scale wind turbine located at INAOE. It had a GoPro camera attached to its back to record video of the inspection, operating with an inverted orientation.

Figure 11: Frontal camera.

Figure 12: GoPro on the back.

The following photographs show the detection of the wind turbine, proceeded by the approach to the hub and inspection of the blades. The flight plan considers the coordinate of the hub first, an inspection point in the middle of the blade and another one at the end, returning to the hub after each blade. After the execution of the trajectory is completed, the UAV lands.

(a) Real setting - Hub.

(b) Real setting - Left blade.

(c) Real setting - Tip of left blade.

(d) Real setting - Right blade.

(e) Real setting - Tip of Right blade.

(f) Real setting - Top blade.

Figure 13: Wind turbine inspection in a real setting.

5 Conclusions

The increased capacity and expanding installation of wind parks require an inspection method capable of producing high quality and readily available data for inspectors. An UAV offers a close-up view of the surface of the rotor blades, increased safety and better mobility than traditional methods.

Motivated by the above advantages, we have presented a method that does not depend on GPS localization, aiming at carrying out the inspection task in an fully autonomous manner.

Our perception system is capable of detecting the blades and hub in simulation using the Hough transform over the HSV color space, and in a real setting with the Line Segment Detector (LSD) over CIE L*a*b.

This automatic detection generates a plan of waypoints, which are then followed autonomously by the flight controller, thanks to its capacity to locate itself with our metric monocular SLAM system.

Furthermore, we carried out several flight tests with two emulated wind turbines of different sizes, facing strong changes in outdoor illumination and background noise that affected the detection. However, we managed to find the right line detectors, coupled with segmentation over suitable color spaces, for each scene.

As future work, we will develop a detection method that is more robust to illumination changes and can handle different scenarios with minimal parameter tuning. This could offer faster setup times for autonomous wind turbine inspection. We will also carry out tests with full-size wind turbines.

Acknowledgements

This research project has been funded by CONACYT grant 291137 and CONACYT-INEGI project 268528. Special thanks to Oyuki Rojas and Yafeng Wang for their contributions to the experimental design.

References

[1] Raul Mur-Artal and Juan D. Tardos. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 10 2017.

[2] L. Oyuki Rojas-Perez and Jose Martinez-Carranza. Metric monocular SLAM and colour segmentation for multiple obstacle avoidance in autonomous flight. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 234–239. IEEE, oct 2017.

[3] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 11 1986.

[4] Richard O. Duda and Peter E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1 1972.

[5] P.V.C. Hough. Method and means for recognizing complex patterns, 3 1960.

[6] Martin Stokkeland. A Computer Vision Approach for Autonomous Wind Turbine Inspection using a Multicopter. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, page 106. Norwegian University of Science and Technology, June 2014.

[7] R.G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, apr 2010.

[8] J Matas, C Galambos, J Kittler, and Y Cvssp. Progressive Probabilistic Hough Transform.

[9] Martin Stokkeland, Kristian Klausen, and Tor A. Johansen. Autonomous visual navigation of Unmanned Aerial Vehicle for wind turbine inspection. Master's thesis, 6 2015.

[10] Sondre Høglund. Autonomous Inspection of Wind Turbines and Buildings using an UAV. Master's thesis, Norwegian University of Science and Technology, 2014.

[11] Bjorn E. Schafer, Davide Picchi, Thomas Engelhardt, and Dirk Abel. Multicopter unmanned aerial vehicle for automated inspection of wind turbines. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 244–249. IEEE, jun 2016.

[12] Huiyi Zhang. A feasibility study of a computer-based wind turbine blades surface flaws inspection method.

Optical-flow-based Stabilization of Micro Air Vehicles Without Scaling Sensors

T.I. Braber,* C. De Wagter, G.C.H.E. de Croon, and R. Babuška
Delft University of Technology, Kluyverweg 1, 2629HS Delft, the Netherlands

ABSTRACT

This article presents an adaptive control strategy to stabilize a micro quadrotor in all three axes using only an Inertial Measurement Unit (IMU) for the attitude control and a monocular camera for canceling position drift. The proposed control scheme automatically determines the appropriate optical flow control gains. This is achieved by extending the stability-based approach to distance estimation developed in [1] to allow for the control of all three axes of a quadrotor. An analysis is done in simulation to present a proof of concept of the stabilization method and to determine the effects of scaling. Furthermore we verify the effects of varying effective camera frame rates and investigate how this control approach generalizes to smaller drone sizes. Actual flight tests are then performed on a Parrot ARDrone 2.0 and on a Parrot Bebop to show that both quadrotors achieve stable hover without position drift using only their IMU and bottom camera.

1 Introduction

Quadrotors are popular due to their simple structure and their combined capability of hovering and performing aggressive maneuvers. Being naturally unstable platforms, quadrotors require accurate and frequent estimations of their attitude, velocity and position. Attitude is obtained through onboard inertial measurements, but velocity and position require extra sensors. The most common outdoor solution is Global Positioning System (GPS) and the most common indoor solution is the use of a camera based external tracking system.

This has enabled impressive feats like pole [2] and ball [3] juggling quadrotors, quadrotors performing aggressive maneuvers [4], flying in swarms [5] and constructing structures [6]. However, in order to be of use in our daily lives, quadrotors also need to be able to fly in unknown and uncontrolled environments. While Simultaneous Localization And Mapping (SLAM) algorithms are widely used to solve these challenges, they are computationally expensive, which highly limits their applicability to smaller quadrotors. Therefore, several researchers are focusing on down-scalable solutions. The authors of [7] used a bottom facing camera combined with ultrasonic height sensor to estimate the quadrotor velocity. Various adaptations of the optic flow algorithms have been made to improve computational efficiency of computing ventral flow and divergence [8]. To counter long term drift in optic flow based velocity control, Li et al. [9] proposed a snapshot-based sensing and control method. To further reduce the required amount of sensors, the approaches in [1] and [10] use a bio-inspired detection of landing height solely based on optic flow divergence and IMU.

Autonomous quadrotors have become smaller with the recent advances in technology, allowing them to be used indoors while being inherently safer to operate around humans. Continuing to scale quadrotors down, however, is not evident as smaller mechanisms are more complex to produce, and parts such as motors cannot be scaled down indefinitely without losing performance. Furthermore the load carrying capability and onboard power of smaller quadrotors is limited, leading to restricted availability of sensors, computational power and flight time.

From [1] it is clear that controlling a quadrotor using only a minimal set of sensors, i.e. a monocular camera and an IMU is possible. This paper presents an extension of the stability-based approach of distance estimation to the control of all three axes. Secondly it presents the effects on the algorithm of downscaling a quadrotor.

Section 2 explains the control solution. In Section 3 we develop a simulation model and present the simulation results. Section 4 shows the results obtained in real flight tests. Finally, Section 5 concludes the paper.

2 Stabilization using Optical Flow (OF)

Without high levels of intelligence to recognize known object sizes in scenes, the use of a single camera gives insufficient information to estimate depth directly. Although time-to-impact can be computed from OF directly, it is not able to differentiate between a small movement close to the scene, and a large movement further away from the scene as shown in Figure 1.

This difference is important for control, as in the latter case, the vehicle must slow down more. There are multiple solutions to solving this problem. As the OF is defined as the ratio between velocity over height, one option would be to simply measure the height using a second camera, or a range sensor. This is the most commonly used approach as most quadrotors have an on-board Ultrasonic Sensor (US) [7] or an Infrared (IR) range finder [8].

*Email address(es): TitusBraber@gmail.com

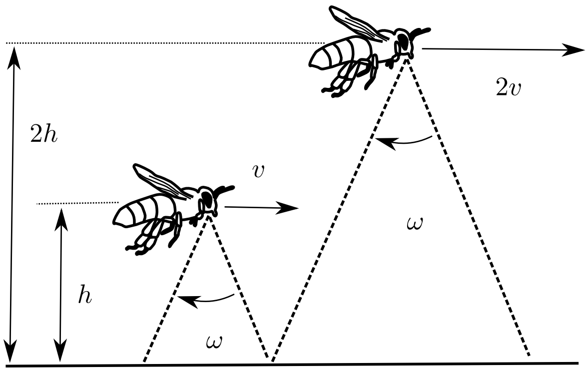


Figure 1: Optical flow does not provide scale.

Recently, different approaches have been proposed to scale optic flow without extra sensors. Given a sufficient acceleration, OF can be scaled based on accelerometers [11]. To control a quadrotor in hover, the need for constant acceleration is not ideal. Similarly, [1] showed that the motion of the quadrotor can be used to estimate the height (appendix B.2), though this also requires an actively moving quadrotor, which can be a problem in indoor environments. Alternatively, [1] also showed that the relationship between the control gain and the height for which the quadrotor becomes unstable due to self-induced oscillations can be used to estimate the height.

Following the proof in [1], there is a linear relationship between the height and the control gain for which the system becomes unstable. This instability leads to oscillations of the quadrotor as the phase shifts between control inputs and visual measurements.

Oscillations can be detected using the Fast Fourier Transform by looking for the specific frequency associated to the oscillations. However, as computational efficiency is of great importance, another method is proposed. Oscillations in the vertical direction can be detected by looking at the covariance of the divergence, which is the vertical speed divided by the height, and efference copies, which are the past thrust control inputs. Alternatively the auto covariance of the divergence can be used to detect a certain frequency of oscillation in the divergence itself, based on the auto covariance delay. However, when the quadrotor would go up and down as a result of an external disturbance such as a wind gust, an oscillation would show up in the auto covariance, as opposed to the covariance of the divergence and efference copies.

When detecting oscillations in the horizontal plane, the thrust is replaced by the effective thrust in the respective axis, or for the sake of simplicity the desired pitch or roll angle. The divergence is replaced by the ventral flow, which is the horizontal speed in X or Y axis divided by the height.

The relationship between gain and height can be used to estimate the current height by increasing the gain until oscillations are detected. With the estimated height, the proper control gains can be set for that height. This procedure can be seen in Algorithm 1 for the detection in the vertical axis.

Algorithm 1 Pseudo code for vertical loop PI control in hover

while true **do**
  if new div from vision module **then**
    $error = -div$
    if $oscillating \neq$ TRUE **then**
      increase gain: $K_p + = a \cdot dt$
      if Cov(div,$thrust$) $> threshold$ **then**
        $oscillating =$ TRUE
        Reduce K_p to stabilize: $K_p = \alpha K_p$
      end if
    end if
    $e_{sum} + = error$
    set $thrust = K_p \cdot error$
  end if
end while

It is worth noting that this method is similar to the “the ultimate gain” PID tuning method by Ziegler and Nichols [12]. This empirical PID tuning method sets the PID gains to zero and increases the P (proportional) gain until the system becomes marginally stable, i.e. begins to oscillate. The P gain for which this happens is called the ultimate gain K_u and the period of the oscillation is T_u. These two values can be used to calculate the gains of the PID controller.

In the case of OF based control however, as the proper gains are height dependent due to the scale invariant measurements from OF, it is not sufficient to perform the Ziegler-Nichols tuning only once. Instead it is done adaptively while flying. Furthermore, the I gain is not set to zero during the period where the P gain is increased as the quadrotor can already substantially drift away with $K_i = 0$.

The resulting system not only tunes the OF based control loop, but at the same time scales the OF measurement.

3 Effects of scaling laws - simulation results

In this section the effects of scaling down a quadrotor will be investigated using simulation. An analysis of the scaling laws is used to properly scale down the Equations of Motion (EoM) used in the simulation. Finally, the effects of varying the Frames per Second (FPS) will be shown.

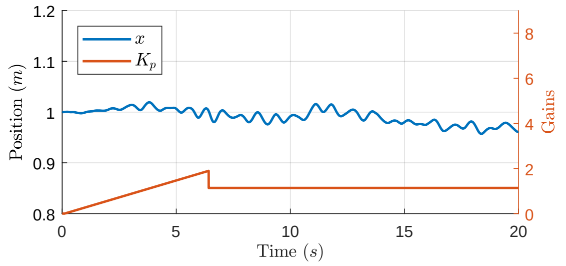
3.1 Simulation model

For simplicity and graphical representation, the simulation will be limited to 2 dimensions: the vertical Z dimension and the horizontal X dimension. The quadrotor is modeled as a rigid body with uniform mass, while the motors are modeled as discretized thrust forces with added Zero Mean White Noise (ZMWN). It is assumed that the maximum thrust the motors can deliver is twice the quadrotors weight. The EoM are as follows:

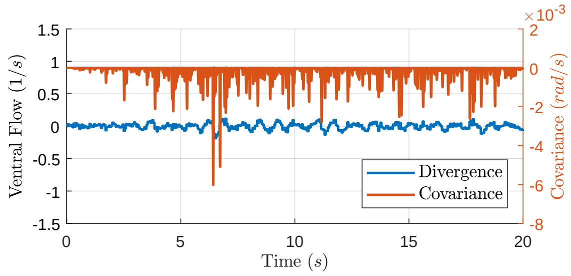
$$\begin{aligned}\ddot{x} &= \frac{F}{m}\sin(\theta)\\ \ddot{z} &= \frac{F}{m}\cos(\theta) - g\\ \ddot{\theta} &= \frac{M}{I}\end{aligned} \tag{1}$$

To simulate the OF measurement, the horizontal and vertical velocity will be taken divided by the height. This represents the ventral flow ($\dot{x}/z$) and divergence($\dot{z}/z$). The velocity and height measurements from the state are not known to the control system. While a typical camera has an update frequency of $30Hz$ or 30 FPS, the time spent computing the ventral flow and divergence might take longer than $1/30$ sec, resulting in a lower *effective* FPS. Therefore, to prevent unrealistic results, the vision signal will have to be discretized by a Zero Order Hold (ZOH) and delayed by a unit delay to model this effective FPS. During each simulation, the effective FPS is fixed, but it will be varied between simulations to study the effect of computational weight of vision algorithms. ZMWN is added to the vision signals before the ZOH, representing the estimation errors made by the vision algorithms.

In Figure 2a the position and the increasing gain in the horizontal axis can be seen with a similar algorithm to Algorithm 1 applied. Oscillations are detected using the covariance in Figure 2b, resulting in a stabilization of the gain.



(a) The position and gain of the quadrotor



(b) The height of the quadrotor for different scales

Figure 2: Results when **not** compensating for scaling

3.2 *Scaling analysis*

The starting point for the simulations will be a quadrotor based on the ARDrone 2, which is further defined as scale 1. This quadrotor will be scaled uniformly, which implies that all parts scale and their properties scale uniformly too. In reality these parts don't scale uniformly however. Instead, different types of motors, batteries, cameras, etc will have to be used, probably leading to differences not reflected in the simulations. The trend however should be visible.

Volume scales with a factor L^3. Mass therefore, assuming constant density, scales with L^3. Moment of inertia scales with a factor L^5, as $I = \int r^2 dm$ with $r \propto L$ and $m \propto L^3$.

The thrust that is generated by the motors, also has to be scaled. According to momentum theory or blade element theory, a thrust F can be approximated during hover in the following form:

$$F = 2\rho \cdot A \cdot v^2 \tag{2}$$

When scaling F in function of L, the density of air is unaffected $\rho \propto 1$, while the surface of the propeller scales with $A \propto L^2$. The rotor tip velocity v, scales differently depending on the assumption of compressibility of the flow[13].

Before making assumptions regarding rotor tip velocity, the effect on forces and moments thus far can be noted as

$$\begin{aligned}F &\propto L^2 \cdot v^2\\ M &\propto F \cdot L \propto L^3 \cdot v^2\end{aligned} \tag{3}$$

In the case of Mach scaling, the flow is assumed to be compressible and the rotor tip velocity to be constant, $v \propto 1$, as opposed to Froude scaling where the flow is assumed incompressible with a constant Froude number. Mach scaling is used here due to the compressibility of air, leading to:

$$\begin{aligned}F &\propto L^2\\ M &\propto L^3\end{aligned} \tag{4}$$

The effect of scaling on linear acceleration, $\ddot{x}, \ddot{z}$ with $F = m \cdot a$ and angular acceleration, $\ddot{\theta}$ with $M = I \cdot \alpha$ can be noted as

$$\begin{aligned}\ddot{x}, \ddot{z} &\propto \frac{L^2}{L^3} \propto L^{-1}\\ \ddot{\theta} &\propto \frac{L^3}{L^5} \propto L^{-2}\end{aligned} \tag{5}$$

3.3 *Scaling compensation*

With the result of the previous section, Equation 1 can be updated in the following way:

$$\begin{aligned}\ddot{x} &= \frac{F \cdot L^2}{m \cdot L^3}\sin(\theta) &&= \frac{1}{L} \cdot \frac{F}{m}\sin(\theta)\\ \ddot{z} &= \frac{F \cdot L^2}{m \cdot L^3}\cos(\theta) - g &&= \frac{1}{L} \cdot \frac{F}{m}\cos(\theta) - g\\ \ddot{\theta} &= \frac{M \cdot L^3}{I \cdot L^5} &&= \frac{1}{L^2} \cdot \frac{M}{I}\end{aligned} \tag{6}$$

It can be seen that smaller quadrotors have significantly faster dynamics due to the scaling in both linear and especially angular accelerations. To prevent the scaling effects from changing the behavior of the quadrotor, the control parameters can be scaled in such a way that they cancel the scaling effects where possible. Therefore the thrust F and moment M should be inversely scaled in comparison to Equation 4. To achieve this, the control parameters K_p, K_i, K_d should be scaled with L and L^2 respectively, looking at the control equation for the thrust and the moment respectively.

The scaling is summarized in Table 1. Please note that only the forces and moments will differ depending on the compensation.

Table 1: Scaling for physical quantities

Parameter	Symbol	Un-	Compensated
Width, Height	w, h	L^1	L^1
Volume, Mass	V, m	L^3	L^3
Moment of Inertia	I	L^5	L^5
Forces	F	L^2	L^3
Moments	M	L^3	L^5

For the simulations, the quadrotor will start from stable hover at $t = 0s$ when the simulation will be started. The control gains will then be increased until oscillations are detected. Once oscillations are detected, the best gain is known and selected, as will be explained further. The results from a simulation with $scale = 1$ form a base result and can be seen in Figure 3. It can be seen that the quadrotor controls its position and altitude for 20 seconds and only very slowly drift centimeters away from its original position.

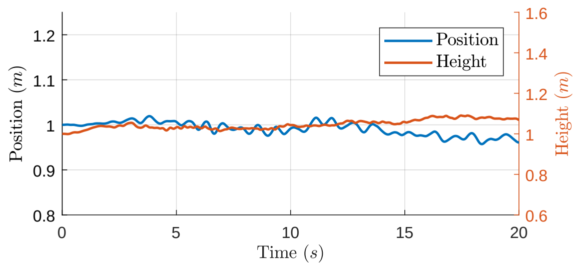


Figure 3: The position and height for $scale = 1$ in simulation

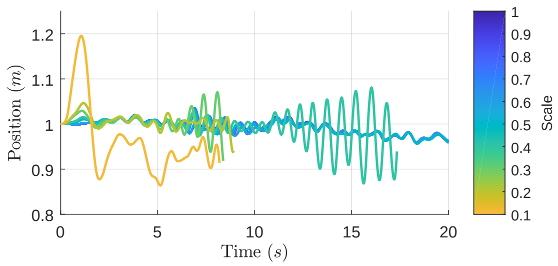
3.3.1 Uncompensated scaling

First, scaling will be applied to the quadrotor without the compensation in control gains mentioned in Section 3.3.

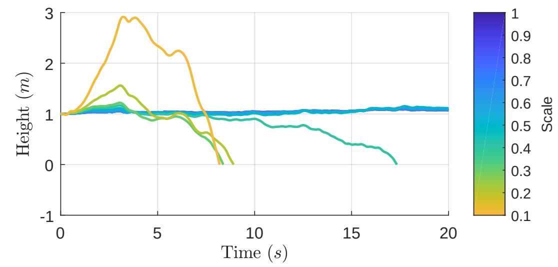
In Figure 4, the height and position of the quadrotor can be seen in the uncompensated case. The scale is varied from 1.0 to 0.1 in steps of 0.1 and the corresponding results are plotted in a color scale ranging from blue to yellow respectively.

Looking at the scales from 0.1 till 0.4 in Figure 4b, it can be seen that the smaller quadrotors become unstable and drift away increasingly as the size goes down. Note that the simulation stops for a scale when the quadrotor touches the ground. The cooler colors representing scales ranging from 0.5 to 1 show stable behavior.

Similarly, in the horizontal axis shown in Figure 4a, we can observe that the smaller drones start to oscillate in the X direction, whereas the larger quadrotors show the same stable behavior as in Figure 3.



(a) The position of the quadrotor for different scales



(b) The height of the quadrotor for different scales

Figure 4: Results when **not** compensating for scaling

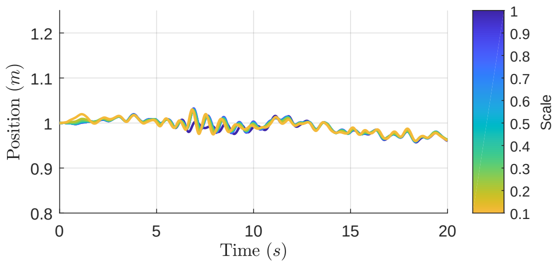
3.3.2 Compensated scaling

When the control forces and moments are scaled with L^3 and L^5 respectively (See Table 1), the plots in Figure 5 show a different result. It can be seen in both the horizontal and vertical axis that all scales are showing stable hover as in Figure 3.

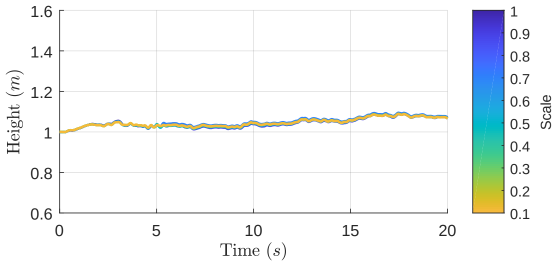
However, in Figure 5, the actuator noise from Section 3 has been scaled too. But in real life, noise tends to increase with smaller scale instead of decrease. To compensate for this increased sensor noise at smaller quadrotor scales, the sensor noise parameter is scaled back with L^2. This yields an important difference, which is shown in Figure 6.

For scale 0.1, the quadrotor goes left and right uncontrollably in the horizontal axis. While the other scales remain stable, there is nevertheless a difference with Figure 5 as the plots no longer overlap. Instead, they all follow the same trend with an increasing amplitude.

Figures 5 and 6 show that the control based on optic flow and IMU in principle can be scaled down to very small sized flying vehicles. The necessary condition, however, is that the

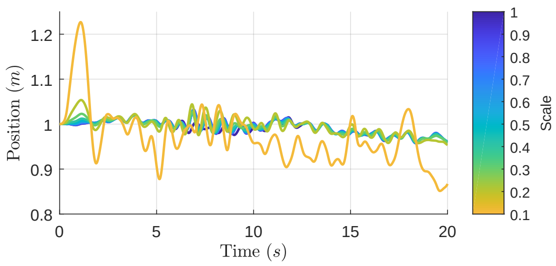


(a) The position of the quadrotor for different scales

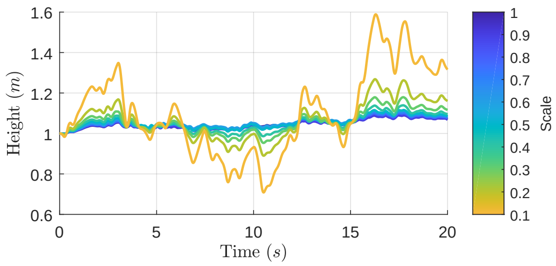


(b) The height of the quadrotor for different scales

Figure 5: Results when compensating for **both** scaling and noise



(a) The position of the quadrotor for different scales

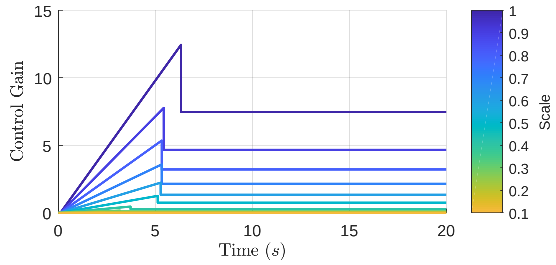


(b) The height of the quadrotor for different scales

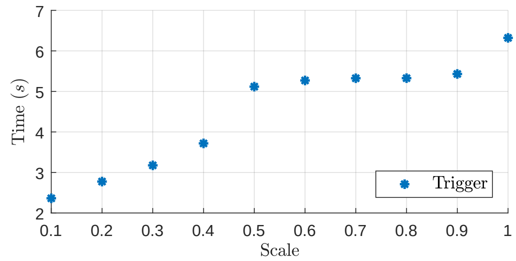
Figure 6: Results when compensating **only** for scaling, **not** for noise

sensor and actuator noise should reduce accordingly. Since in practice this is not the case, a minimal practical size exists.

As shown explained in detail in [1], the control gain of optic flow based control depends on the distance the observed surface. By slowly increasing the gain until oscillation is detected, the optimal gain is found. Figure 7 shows this gain



(a) The vertical gains of the quadrotor for different scales



(b) The vertical trigger points of the quadrotor for different scales

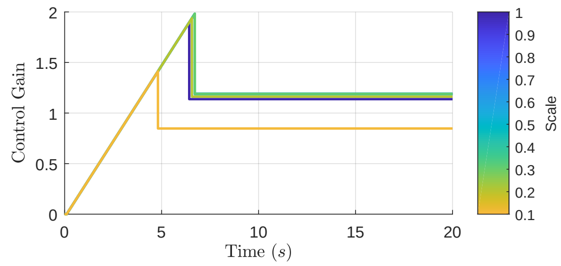
Figure 7: Vertical axis when compensating **only** for scaling, **not** for noise

increase and shows when the quadrotor detects a starting oscillation and selects the final gain. The increasing gains of the vertical axis are depicted in Figure 7a. The algorithm is triggered at roughly the same time for scales 0.9 through 0.5 between 5.4 to 5.1 seconds, while the gains from 0.4 to 0.1 trigger from 3.7 seconds to 2.4 seconds. Furthermore, scale 1 triggers at 6.3 seconds. Alternatively when plotting the scale against the trigger time, as Figure 7b shows, one could see a linear relation, with scales 0.9 to 0.5 breaking this trend.

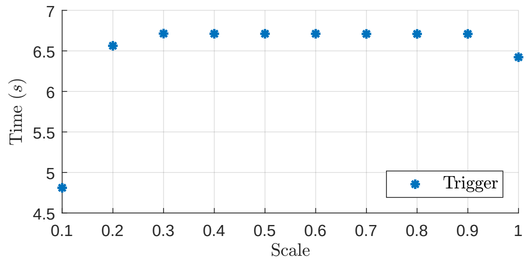
Looking at the horizontal axis in Figure 8 however, the triggers are all around the same time, 6.6 seconds, except for scale 0.1, which trigger at 4.8 seconds. This might again be explained by the fact that the actuator noise works in the thrust direction of the quadrotor, which is mainly vertical while hovering. The fact that all scales find roughly the same gain is because in the horizontal loop, the desired pitch angle θ is the actuator, which is in radians and thus not influenced by scaling.

3.4 *Influence of varying effective FPS*

The effect of reducing the FPS is analyzed using the model with scale$= 1$. The quadrotor is set to hover in simulation at initial position in the horizontal axis of 1 meter and height 1 meter. Next the adaptive gain algorithm is started in both axes for different effective FPS. Figure 9 shows the response of the quadrotor in function of decreasing FPS. Below 15 FPS, the control of the quadrotor becomes unreliable and either starts to oscillate in the horizontal direction which also causes the drift in vertical direction or even becomes unstable at even lower frame rates.

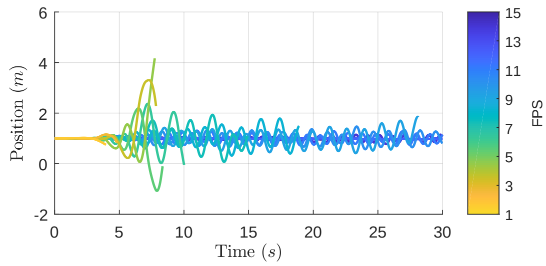


(a) The horizontal gains of the quadrotor for different scales

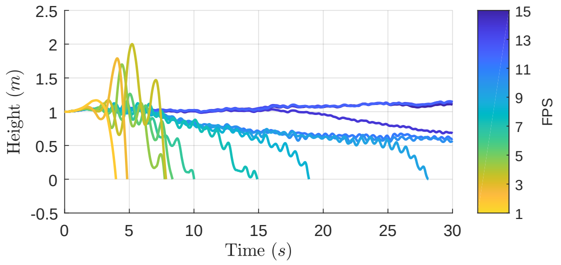


(b) The horizontal trigger points of the quadrotor for different scales

Figure 8: Horizontal axis when compensating **only** for scaling, **not** for noise



(a) The position of the quadrotor for different effective FPS



(b) The height of the quadrotor for different effective FPS

Figure 9: Determining the minimal FPS to fly a quadrotor of scale 1 using the adaptive gain strategy in both axis

We also investigate if it is possible to stabilize the quadrotor at smaller scales, as seen in Figure 6, with a higher effective FPS. The same experiment is performed, but for the smaller scales and higher effective FPS. Figure 10 shows the height of a 0.2 quadrotor, with an FPS scale from the previously used 20 FPS in yellow, to 100 FPS in blue, with steps

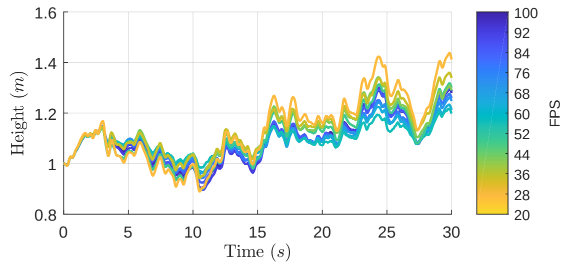


Figure 10: The height of a 0.2 scale quadrotor with increasing FPS

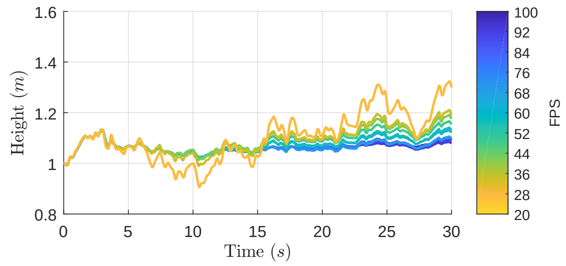


Figure 11: The height of a 0.2 scale quadrotor with increasing FPS without noise on the vision

of 8 FPS. Even though the quadrotor seems to benefit from a higher FPS, it is still not as stable as the 1 scale quadrotor was at 20 FPS. Furthermore, it seems there is an optimum, in this case 52 FPS. Figure 11 shows the same experiment, but without noise on the vision. Not only is this flight more stable than with noise, but now the increasing frame rate results in increasingly stable flight.

From this analysis, it can be seen that higher FPS only clearly helps if the noise is sufficient low, and that at lower scales even higher FPS can not nicely stabilize the quadrotor.

4 Flying a quadrotor - real life results

This section presents a complete control strategy that achieves stable hover in an unknown environment using just a single camera and an IMU. In the simulation results presented in Section 3 the algorithm was applied to both axes at the same time. With knowledge of the gain height relationships in all three axes however it is sufficient to use one axis to estimate the height and set the proper control gains for all three axes. This would require the quadrotor to apply Algorithm 1 in a single axis only, preferably the vertical one as it is the most stable one. Prior to this the linear gain height relationships should be determined for all axes.

4.1 Finding the gain height relationships

To determine the gain height relationships a quadrotor is flown in a controlled environment, the CyberZoo at the faculty of Aerospace Engineering at Delft University of Technology, where an OptiTrack motion capture system is installed to acquire ground truth measurements. The quadrotor is set to hover and the algorithm is started in a single axis, while the other two axes are controlled using the ground truth measurements. When the quadrotor detects an oscillation it triggers to note the gain and the ground truth height. This experiment is repeated multiple times at several heights for all axes, allowing a linear fit through the data.

An ARDrone 2 flying with the open source Paparazzi software was used. The bottom camera acquired the ventral flows and divergence using the Lucas-Kanade (LK) algorithm at an effective FPS between 20 and 30. A picture of the experiment can be seen in Figure 12.

Figure 12: The ARDrone 2 during an experiment

It was assumed that the X and Y axes were symmetrical enough to determine the gain height relationship only in the Y axis. The results can be seen in Figure 13 for the horizontal axes and Figure 14 for the vertical axis.

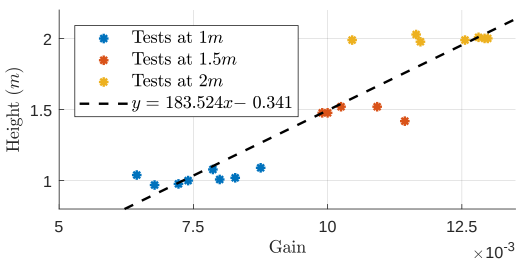


Figure 13: The gain and height for each trigger point in the horizontal axis

The fit for the horizontal axis shows a linear relationship for the gain and the height, at $y = 183.524x - 0.341$, and the vertical axis at $y = 0.995x + 0.066$.

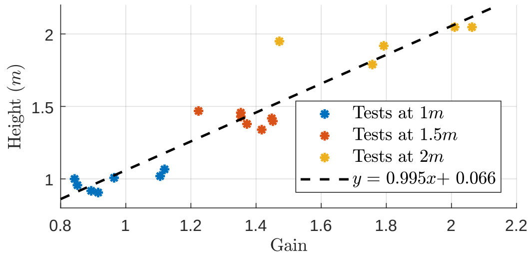


Figure 14: The gain and height for each trigger point in the vertical axis

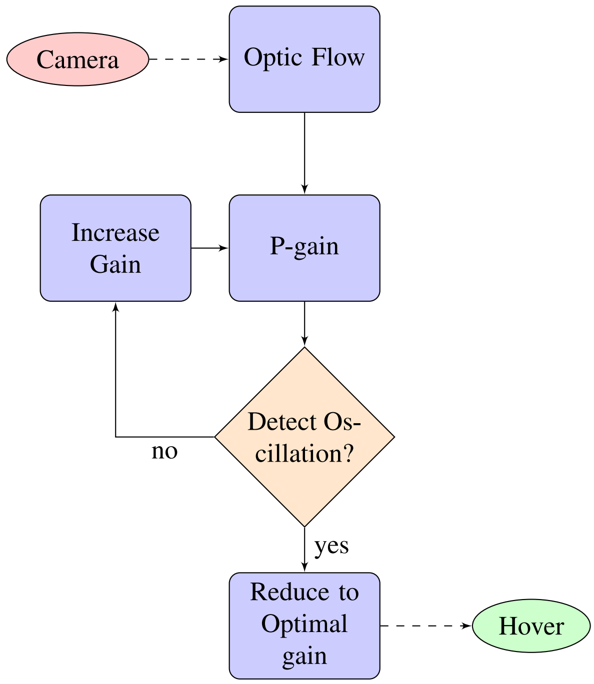
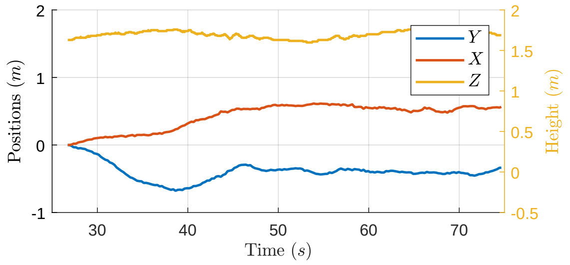


Figure 15: Flow chart of the optic flow based control

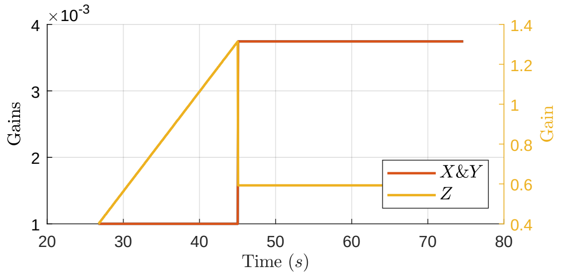
4.2 Stable hover in an unknown environment

To show the validity of this approach the ARDrone 2 is set to hover at an unknown height and the algorithm (See Figure 15) is started in the vertical axis, this can be seen in Figure 16b. Meanwhile the quadrotor is drifting away in the horizontal X and Y axes, as can be seen in Figure 16a. When the covariance of the divergence and thrust inputs dives below a threshold in Figure 16c the quadrotor detects the start of an oscillation it estimates the height it is flying at and and sets the proper gains in all three axes, using the gain height relationships as determined in Section 4.1.

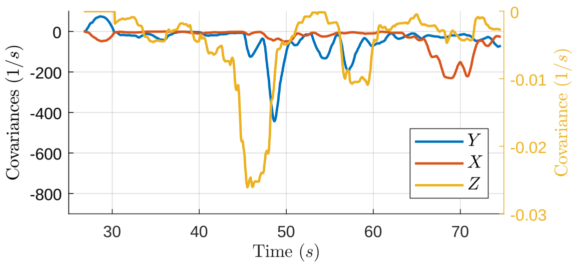
It is clearly visible that the trend of drifting in the horizontal direction is stopped and the quadrotor stabilizes around the position it was in when the proper gains were selected.



(a) The X and Y position and the height of the ARDrone 2



(b) The gains in X, Y and Z



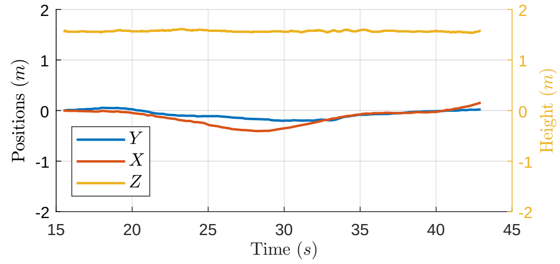
(c) The covariances used as a trigger

Figure 16: The ARDrone 2 with the algorithm applied to the vertical axis, to set proper gains in all axes using relationships between height and gain

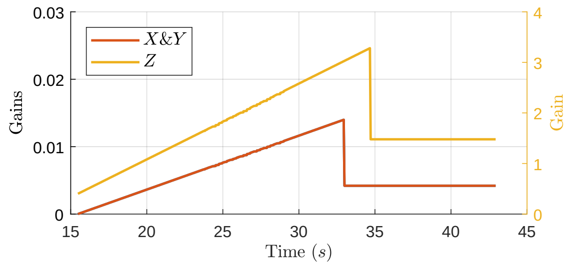
This experiment, amongst others, can also be seen in the following Youtube play list[1]. In video *Ardrone 2: Drift* the experiment mentioned can be seen, while video *Ardrone 2: drag* shows that the quadrotor stays above the texture mat, even if it is dragged around the Cyber Zoo.

Furthermore the video *Bebop: Simultaneous* shows an alternative approaches to the control strategy that will also work when no gain height relationship has been determined previously on the quadrotor. The video shows a Bebop quadrotor applying the algorithm on both the vertical axis and the horizontal axes at the same time. The same experiment is can be seen in Figure 17. When oscillations are detected in the vertical axis, the respective gain is lowered to stabilize that axis. When oscillations are detected in one of the horizontal axes, both gains are lowered to stabilize the horizontal axes.

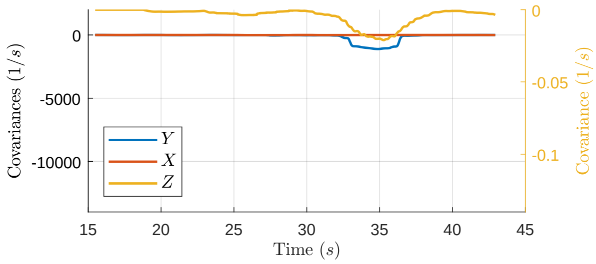
All code is released open-source in the Paparazzi-UAV project[2]



(a) The X and Y position and the height of the Bebop



(b) The gains in X, Y and Z



(c) The covariances used as a trigger

Figure 17: The Bebop with the algorithm applied to the all axes at the same time. A trigger in one horizontal axis will set the gain for both horizontal axes

[1] `https://www.youtube.com/playlist?list=PL_KSX9GOn2P-Ire2SSVgxLnZorQ6jbgN1`

[2] `https://github.com/paparazzi/paparazzi/blob/master/conf/modules/optical_flow_hover.xml`

5 Conclusion

In this article we have shown a novel control strategy to control a micro quadrotor drift free in an unknown environment using only an IMU and a monocular camera. The algorithm adaptively selects the proper control gains for the estimated height, using a stability-based approach to estimate distance. This was achieved by extending [1] to all axes and using the vertical height estimation to set control gains for both the vertical and horizontal axes from the predetermined gain height relationships. Alternatively the control strategy can also be used without predetermining these relationships, by applying the algorithm in each axes separately.

First the effects of scaling on stabilization of micro quadrotors have been shown by simulation. Though the dynamics are significantly faster for smaller quadrotors, especially the in the rotational degrees of freedom, it can be concluded that by proper scaling of the control gains most of the effects can be compensated for. The effects of noise however cannot be compensated for by inversely scaling control parameters when scaling down, leaving an increased effect of noise on the smaller quadrotors. It was also demonstrated that increasing the effective FPS would help decrease these effects on smaller quadrotors.

Finally it was successfully shown that quadrotors could hover without drifting away in both simulation and on an Parrot ARDrone 2 and Bebop using various versions of the control strategy.

Acknowledgements

This article is the condensation of my MSc thesis *Vision-based stabilization of micro quadrotors*, September 2017. I would like to thank my supervisors and fellow authors of this article again for their help and support during my thesis.

References

[1] Guido C.H.E. de Croon. Monocular distance estimation with optical flow maneuvers and efference copies: a stability-based strategy. *Bioinspiration & Biomimetics*, 11(1):016004, 2016.

[2] Dario Brescianini, Markus Hehn, and Raffaello D'Andrea. Quadrocopter pole acrobatics. *IEEE International Conference on Intelligent Robots and Systems*, pages 3472–3479, 2013.

[3] Mark Müller, Sergei Lupashin, and Raffaello D'Andrea. Quadrocopter ball juggling. *IEEE International Conference on Intelligent Robots and Systems*, pages 5113–5120, 2011.

[4] Daniel Mellinger, Nathan Michael, and Vijay Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *Springer Tracts in Advanced Robotics*, 79:361–373, 2014.

[5] Alex Kushleyev, Daniel Mellinger, Caitlin Powers, and Vijay Kumar. Towards a swarm of agile micro quadrotors. *Autonomous Robots*, 35(4):287–300, 2013.

[6] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. Construction of Cubic Structures with Quadrotor Teams. *Mechanical Engineering*, 2011.

[7] Pierre-Jean Bristeau, FranÃ§ois Callou, David VissiÃ¨re, and Nicolas Petit. The navigation and control technology inside the ar.drone micro uav. *IFAC Proceedings Volumes*, 44(1):1477 – 1484, 2011. 18th IFAC World Congress.

[8] K N McGuire, G.C.H.E. de Croon, Christophe. De Wagter, K Remes B.D.W. Tuyls, and H J Kappen. Local Histogram Matching for Efficient Optical Flow Computation Applied to Velocity Estimation on Pocket Drones. *In press*, pages 1–15, 2015.

[9] Ping Li, Matthew Garratt, and Andrew Lambert. Monocular Snapshot-based Sensing and Control of Hover, Takeoff, and Landing for a Low-cost Quadrotor. *Journal of Field Robotics*, 32(7):984–1003, 2015.

[10] H. W. Ho, G. C. H. E. de Croon, E. van Kampen, Q. P. Chu, and M. Mulder. Adaptive gain control strategy for constant optical flow divergence landing. *IEEE Transactions on Robotics*, 34(2):508–516, April 2018.

[11] Volker Grabe, Heinrich H Bulthoff, and Paolo Robuffo Giordano. A comparison of scale estimation schemes for a quadrotor uav based on optical flow and imu measurements. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5193–5200. IEEE, 2013.

[12] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *InTech*, 42(6):94–100, 1995.

[13] Chester H Wolowicz and James S Bowman. NASA Technical Paper 1435 Similitude Requirements and Scaling Relationships as Applied to Model Testing NASA Technical Paper 14-35 Similitude Requirements and Scaling Relationships as Applied to Model Testing. (August):65, 1979.

Real-time Disparity Map Reconstruction with On-board FPGA by Semi-global Matching and Weighted Least Square Filtering

Pengfei Wang[1],Zhi Gao[1,*],Hailong Qin[2], Myo Tun Aung[1],
Xudong Cheng[3], Mingjie Lao[1], Feng Lin[1], Swee Huat Rodney Teo[1]
1. Temasek Laboratories, National University of Singapore
2. Roadstar.ai
3. Carnegie Mellon University

ABSTRACT

In this paper, we propose a real-time disparity map estimation framework on FPGA, combined with an effective post-processing method. Given the input stream of stereo image pairs, a semi-global matching based framework is implemented on the FPGA to estimate the disparity in real-time. The generated disparity map is refined with a weighted least square (WLS) filtering method. The experiments show that the disparity map can be reconstructed in real-time. In addition, the weighted least square filtering based post-processing can significantly improve the accuracy of the disparity map and remove large estimation errors.

1 INTRODUCTION

Stereo Vision, resulting in the knowledge of depth information in a scene, is of great importance in the field of robotics vision. Generally speaking, if the two images are rectified, matching pixels reside on corresponding horizontal lines. For each pixel $p_l(x, y)$ in the left image, its possible matching pixel is $p_r(x-d, y)$, where d is the disparity for the matched pixels. The searching range of disparity is [0, d_{max}], where d_{max} is the maximum disparity, as shown in Fig. 1. The disparity map is composed by the disparity value. Once we get the disparity map, the depth of scenery can be calculated by a simple equation $z = fb/d$, where b is the baseline, f is the focal length. The algorithms for disparity estimation can be generally classified into three categories according to the matching cost, local method, global method and semi-global method. Block matching is one of the typical kind of local method, the matching cost is a summation of the difference within a small neighbourhood of each pixel in the left and right image. Local methods are often sensitive to noise. Global methods treat disparity estimation as a multi-label problem and construct a 2D graph optimized by algorithms of graph cut [1] or belief propagation [2, 3, 4]. Semi-global methods [5] are also used to approximate the NP-hard 2D graph as independent scan-lines and leverages dynamic programming to aggregate the matching cost. Global algorithms typically do not perform an aggregation step, but rather seek a disparity assignment (step 3) that minimizes a global cost function that combines data (step 1) and smoothness terms. While the 2D-optimization can be shown to be NP-hard for common classes of smoothness functions [6], dynamic programming can find the global minimum for independent scan-lines in polynomial time. Semi-global method is our choice for disparity estimation as it has both high computation efficiency and accuracy, compared to local methods and global methods.

Nowadays, there exists many algorithms to build accurate correspondence between a pair of images. Besides, dedicated hardware platforms such as FPGAs and GPUs should be utilized to realize real-time processing through speeding up stereo vision systems. Some researches focus on stereo matching algorithms which use local methods or semi-global methods accelerated by GPUs [7]. [5] reached 12 fps at 450 $\times$ 375 resolution with 64 disparities and [8] achieved 8 fps for 320 $\times$ 240 pixel images with 64 disparity levels. FPGA costs less power and its memory hierarchies as well as processing units could be configured according to user needs [9], thus FPGA outperforms GPU to some extents. [10] achieved 60 fps at 1024 $\times$ 768 pixel stereo images by merging cross-based cost aggregation and mini-census transform. In [11], mini-census adaptive support region stereo matching algorithm was applied and the experiments in this paper shown that 47.6 fps for 1920 $\times$ 1080 with a disparity range of 256 and 129 fps for 1024 $\times$ 768 with 128 disparity could be attained respectively. [9] combined cost aggregation and fast locally consistent dense stereo methods. By combining the two aforementioned methods and testing on Xilinx Virtex-6 FPGAs, it reached 507.9 fps for 640 $\times$ 480 pixel images. Although this paper got low error rate with high frequency, it could not obtain good results with high-definition image. 1600 $\times$ 1200 pixel images with 128 disparity levels at 42 fps was achieved in [12], whose test results were evaluated on Altera Stratix-V FPGAs.

*Email address(es): contact_gaozhinus@gmail.com

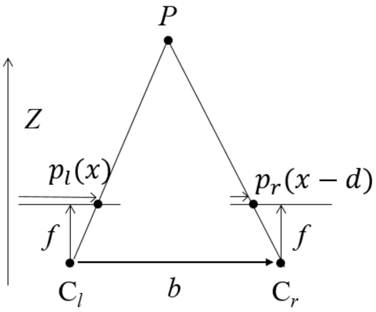


Figure 1: Disparity

2 Proposed Method

Considering the balance between accuracy and efficiency, we take semi-global method [5] as our framework for disparity estimation. The problem of disparity estimation can be modelled as an energy minimization problem, as shown in Equation 1.

$$E(D) = \sum_{p}(C(\mathbf{p}, D_p) + \sum_{q \in N_p} P_1 T[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1]) \tag{1}$$

where, N_p is the neighbour of p, and P_1 and P_2 is the penalty for disparity variation for neighbouring pixels. The energy consists of pixel-wise matching cost, smoothness cost, and edge-preserving cost. This is an NP-hard problem, and it can be approximated by minimization along 4 individual scan-lines with dynamic programming. We also add a WLS filter as the post-processing to improve the performance after the disparity map is estimated. The whole algorithm consists of the following steps:

- Median filtering is applied to filter the speckle noise such as salt and pepper noise.
- Matching cost for each pixel is calculated by census transform and hamming distance.
- The cost of the pixels are aggregated along scan-lines with dynamic programming.
- For each pixel, the disparity with the smallest matching cost is selected with a winner-take-all strategy.
- Weighted least square filter based post-processing is applied to improve the result of disparity matching.

All the steps except the post-processing are implemented on FPGA.

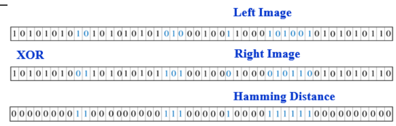


Figure 2: Hamming distance

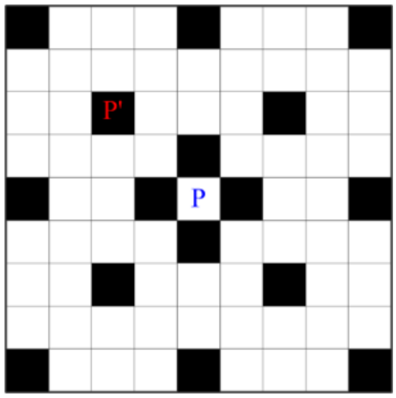


Figure 3: Sparse census transform

2.1 *Algorithm for Disparity Estimation*

The input left and right images are rectified and fed into a 3×3 median filter. Census transform is applied on both the left and right images, as in Equation 2. The matching cost between $p_l(x, y)$ and $p_r(x-d, y)$ is determined as the hamming distance (Fig. 2) between the two binary vectors obtained by census transform, as in Equation 3. In order to enlarge the size of the neighbour without obviously increasing computation time, we implement a sparse census transform pattern with a window of size 9×9, in which only 16 pixels are counted in the census transform (Fig. 3). The matching cost is aggregated as Equation 4. After cost aggregation, the disparity with the minimal cost is selected for each pixel.

$$\xi(p, p') = \begin{Bmatrix} 1, if I(p') < I(p) \\ 0, otherwise \end{Bmatrix} \tag{2}$$

$$C(p, d_p) = \bigotimes_{p' \in N(p)} \xi(p, p') \tag{3}$$

$$\begin{aligned} L_r(p, d) = & C(\mathbf{p}, d) + min(L_r(\mathbf{p} - \mathbf{r}, d), \\ & L_r(\mathbf{p} - \mathbf{r}, d-1) + P_1, \\ & L_r(p - r, d+1) + P_1, \\ & min_i L_r(\mathbf{p} - \mathbf{r}, i) + P_2) \\ & - min_k L_r(\mathbf{p} - \mathbf{r}, k) \end{aligned} \tag{4}$$

2.2 *FPGA Implementation for Disaprity Estimation*

The algorithm of the disparity estimation was implemented on a Xilinx ZYNQ XC7Z045 FPGA. The window size of median filter is 3×3. The window size for the census transform is 9×9. Pixels slide through 9×9 shift register

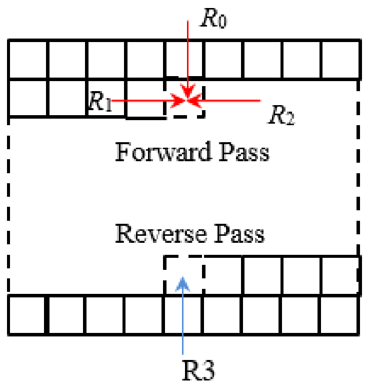


Figure 4: Forward and reverse path of cost aggregation

representing the neighborhood and loaded into census computation block. The Hamming Distance is calculated through an XOR operation followed by count of ones, the summation of set bits. The census transform of left image is stored in a d_{max} 16-bit shift register and XOR with the census transform of the right image. At each clock cycle, cost for all disparity levels are generated and stored in output BRAM.

For the step of cost aggregation, the 4 directions are split into two paths, forward pass and reverse path, as shown in Fig. 4. The cost calculation for each direction is performed using Equation 4. It concurrently computes the aggregation cost for all disparity levels and three paths. The implementation of the elementary block of the aggregation cost computation is shown in Fig. 5. The block is replicated d_{max} times for each 3 paths, resulting $3 \times d_{max}$ blocks to process concurrently. The bottom multiplexer selects the output between the initial cost for the beginning of a path, and the currently computed value along the path.

After the cost aggregation, the disparity with the minimal cost for each pixel is to be selected. A serial tree structure comparator is applied for disparity selection. The implementation uses $d_{max}/2$ comparators with computation complexity of $O(d_{max})$, as shown in Fig. 6.

Fig. 7 shows the time schedule for each module in the processing of one frame. Data load, census transform, and hamming distance are pipe-lined. The generated cost by hamming distance is buffered in external DDR memory. The forward pass starts after DDR buffering. The reverse pass and depth map generation starts after forward pass completed.

2.3 Post-processing by Weighted Least Square Filtering

The accuracy of the disparity estimation is often suffered from extreme scenario, such as texture-less region, overexposure, repetitive structure, etc. In order to improve the accuracy of the disparity, post-processing is always necessary. We take the weighted least square filtering [13] for post-processing, for its good performance of edge preserving smoothing. The objective of filtering the disparity can be

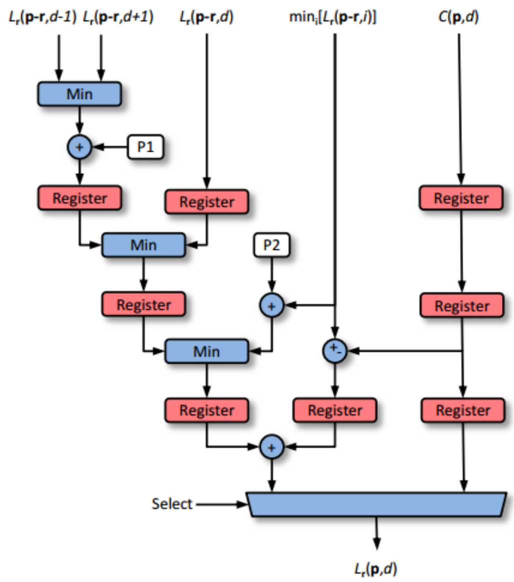


Figure 5: Cost aggregation structure on FPGA

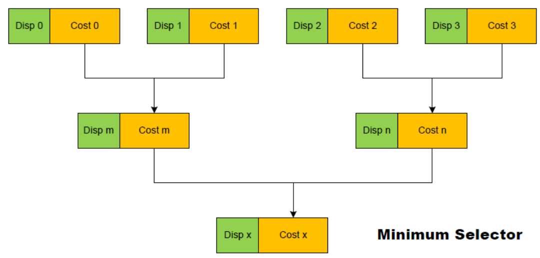


Figure 6: Tree structure comparator for disparity selection

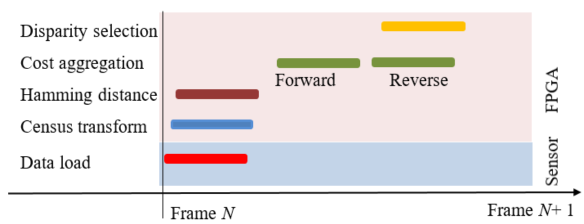


Figure 7: Timing diagram of the system

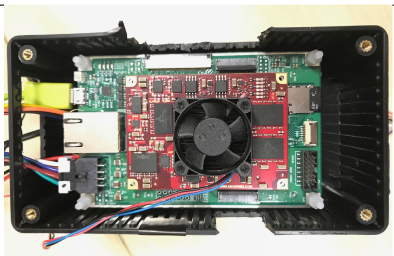


Figure 8: The whole system

expressed as minimizing Equation 5.

$$\sum_p ((D'_p - D_p)^2 + \lambda(a_{x,p}(I)(\frac{\partial D}{\partial x})^2_p + a_{y,p}(I)(\frac{\partial D}{\partial y})^2_p)) \quad (5)$$

$a_{x,p}(I)$ and $a_{y,p}(I)$ are the smoothness weights as defined in Equation 6 as [14]:

$$\begin{aligned} a_{x,p}(I) &= (\left|\frac{\partial l}{\partial x}(p)\right|^{\alpha} + \epsilon)^{-1} \\ a_{y,p}(I) &= (\left|\frac{\partial l}{\partial y}(p)\right|^{\alpha} + \epsilon)^{-1} \end{aligned} \quad (6)$$

where l is the log-luminance channel of the guidance image I, the parameter α determines the edge sharpness, while ϵ is a small constant, for example, 0.0001.

3 Experiments

The disparity map estimation algorithm is implemented on Xilinx ZYNQ ZC7045 FPGA, and the post-processing is implemented on ARM processor. The whole system is shown in Fig. 8.

In order to test the performance of the algorithm, we have carried out experiments on Middlebury dataset and self-collected data. The sample result of the image "Teddy" from the dataset is shown in Fig. 9. The result shows that the WLS filter can improve the accuracy of the estimated disparity.

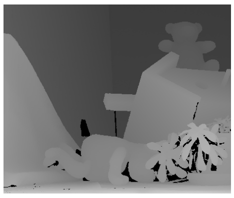
For the self-collected data (Fig. 10), the result of the estimated disparity by semi-global matching suffers stride effect, especially at the right-top corner with overexposure, the disparity is totally wrong as it is a large texture-less region. The weighted least square filter can remove the stride effect, makes the disparity level more clear, and reduce the error caused by the overexposure.

4 Conclusion

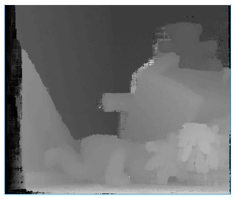
In this paper, we have implemented a Semi-global disparity estimation algorithm on FPGA, and improve the disparity map through weighted least square filtering. In the disparity estimation, the sparse census transform has two folds, large window makes the descriptor more robust to noise, and the computation load increases not too much. The WLS filtering based can improve the accuracy of the estimated disparity

(a) Left image

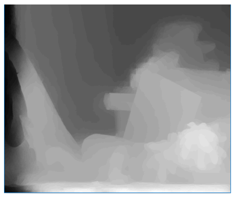
(b) Right image



(c) Disparity groundtruth



(d) Disparity by SGM, error rate=8.0%

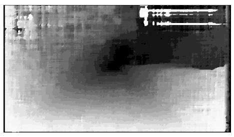


(e) Disparity after WLS, error rate=6.7%

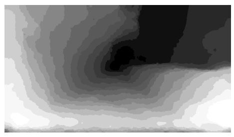
Figure 9: Estimated disparity by SGM and WLS for Middlebury "Teddy"

(a) Left image

(b) Right image



(c) Disparity by SGM



(d) Disparity after WLS

Figure 10: Estimated disparity by SGM and WLS for self-collected data

map, and reduces some defects, such as error caused by overexposure.

References

[1] Michael Bleyer and Margrit Gelautz. Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions. *Signal Processing: Image Communication*, 22(2):127–143, 2007.

[2] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient belief propagation for early vision. *International journal of computer vision*, 70(1):41–54, 2006.

[3] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on pattern analysis and machine intelligence*, 25(7):787–800, 2003.

[4] Qingxiong Yang, Liang Wang, and Narendra Ahuja. A constant-space belief propagation algorithm for stereo matching. In *Computer vision and pattern recognition (CVPR), 2010 IEEE Conference on*, pages 1458–1465. IEEE, 2010.

[5] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2008.

[6] Olga Veksler and Ramin Zabih. Efficient graph-based energy minimization methods in computer vision. 1999.

[7] Ruigang Yang, Greg Welch, and Gary Bishop. Real-time consensus-based scene reconstruction using commodity graphics hardware. In *Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on*, pages 225–234. IEEE, 2002.

[8] Ilya D Rosenberg, Philip L Davidson, Casey MR Muller, and Jefferson Y Han. Real-time stereo vision using semi-global matching on programmable graphics hardware. In *ACM SIGGRAPH 2006 Sketches*, page 89. ACM, 2006.

[9] Minxi Jin and Tsutomu Maruyama. Fast and accurate stereo vision system on fpga. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 7(1):3, 2014.

[10] Lu Zhang, Ke Zhang, Tian Sheuan Chang, Gauthier Lafruit, Georgi Krasimirov Kuzmanov, and Diederik Verkest. Real-time high-definition stereo matching on fpga. In *Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays*, pages 55–64. ACM, 2011.

[11] Yi Shan, Yuchen Hao, Wenqiang Wang, Yu Wang, Xu Chen, Huazhong Yang, and Wayne Luk. Hardware acceleration for an accurate stereo vision system using mini-census adaptive support region. *ACM Transactions on Embedded Computing Systems (TECS)*, 13(4s):132, 2014.

[12] Wenqiang Wang, Jing Yan, Ningyi Xu, Yu Wang, and Feng-Hsiung Hsu. Real-time high-quality stereo vision system in fpga. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(10):1696–1708, 2015.

[13] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM Transactions on Graphics (TOG)*, volume 27, page 67. ACM, 2008.

[14] Dani Lischinski, Zeev Farbman, Matt Uyttendaele, and Richard Szeliski. Interactive local adjustment of tonal values. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 646–653. ACM, 2006.

Obstacle Avoidance for UAVs via Imitation Learning from Human Data with 3D-Convolutional Neural Networks

Bumsoo Park and Hyondong Oh*
School of Mechanical, Aerospace, and Nuclear Engineering, UNIST, Ulsan, South Korea

ABSTRACT

In this paper, we present a novel framework that can be applied for the obstacle avoidance of unmanned aerial vehicles (UAVs) based on deep learning methods with supervision from actual human flight control data. Through imitation learning from human flight control data, the UAV is expected to learn how to avoid obstacles without any given rules, and at the same time learn the intuition that humans possess to efficiently deal with unexpected situations. One critical limitation for UAVs is that the number and size of sensors that can be attached is restrictive, hence a monocular camera will be used as the only sensor of the UAV. The simulation is conducted within a simulated environment using Gazebo and ROS (Robot Operating System), where the visual input from the camera and human control input regarding the direction of the UAV are utilized for the training process. The trained model is then validated in terms of how well it imitates a human and how capable it is to avoid obstacles.

1 INTRODUCTION

The recent upsurge of demand for mobile robots with higher levels of autonomy has posed several challenges especially when it comes to the ability for autonomous navigation within a given environment. Among the requirements for fully autonomous navigation of an unmanned agent, obstacle avoidance is considered one of the most fundamental factors, as the agent should be capable of avoiding obstacles during its navigation. Autonomous navigation is usually implemented through path planning methods, where an agent is assigned a certain path given that it has information on the surrounding environment. In general, geometrical constraints such as obstacles between the start and goal point are considered throughout the path planning stage, facilitating obstacle avoidance for the autonomous agent. However, real-time path planning may be unavailable under certain circumstances where the environment is uncertain, or where the path cannot be generated due to issues such as the inability to obtain a map of the surrounding environment. In such cases, local path planning methods take place instead of global path planning methods [1], where obstacle avoidance is considered as the main criterion. Hence, we will focus on the investigation of obstacle avoidance methodologies as it is one of the most fundamental requirements for a mobile agent to exhibit fully autonomous behaviours.

Conventional real-time obstacle avoidance techniques employ rule-based methodologies, where certain rules are set for the agent according to the sensor input. Such researches include obstacle avoidance methods with the usage of ultrasonic sensors [2, 3], or optical flow-based obstacle avoidance methods for monocular camera usage [4, 5]. These methods, however, may also lose its robustness as the agent can undesirably encounter unexpected situations which are not considered in the rule-based methods. Recent researches include data-driven or machine learning-based approaches that use

*contact email: h.oh@unist.ac.kr

apprenticeship learning methods [6], yet the algorithms that are used for the supervision of the machine learning model are still developed upon human-made rules. Hence, in some occasions, learning directly from human demonstrations may provide better results depending on the task, especially when there are constraints on the usage of sensors. In fact, there have been multiple attempts regarding imitation learning from human experts. One of the most primitive researches was proposed by Pomerleau [7], where a single layer of a neural network was used to map the images to the steering angles, enabling the agent to stay on the road. Other work regarding how to teach an aircraft to fly using human data [8] has been introduced by Sammut, Hurst, Kedzier, and Michie. In this work, decision trees were used to train and design the autopilot for the aircraft. One representative example of imitation learning from human data was presented by Ross, Gordon, and Bagnell [9], using the DAGGER (Dataset Aggregation) algorithm. Here, a policy is determined online in an iterative manner, using the aggregated data from the algorithm. This algorithm was also used for vision-based autonomous navigation in forest trails [10]. Kim and Chen [11] presented an imitation learning framework for drones to track and follow a certain target based on neural networks.

Under the main assumption that there underlies a certain relationship or function between visual inputs and actions of a human when it comes to obstacle avoidance, we focused on teaching an agent so that it can map the image inputs to the heading directions for obstacle avoidance. Through sufficient acquisition of camera data and the corresponding control inputs generated from human experts, our neural network is likely to learn the underlying rule for obstacle avoidance. In this paper, we present a framework for obstacle avoidance with deep learning-based imitation learning methods. 3D-CNN models [12] will be utilized for the imitation learning framework, as obstacle avoidance with a monocular camera is considered a sequential task.

A general description of the background will be given in Section 2. The methodology for the proposed imitation learning framework followed by details on how the neural network model is trained are discussed in Section 3. In Section 4, we will discuss the training results of the neural network, and validate the performance of the obstacle avoidance capabilities by applying the trained neural network model to a simulated drone. Finally, Section 5 includes the conclusion and discussion for this work.

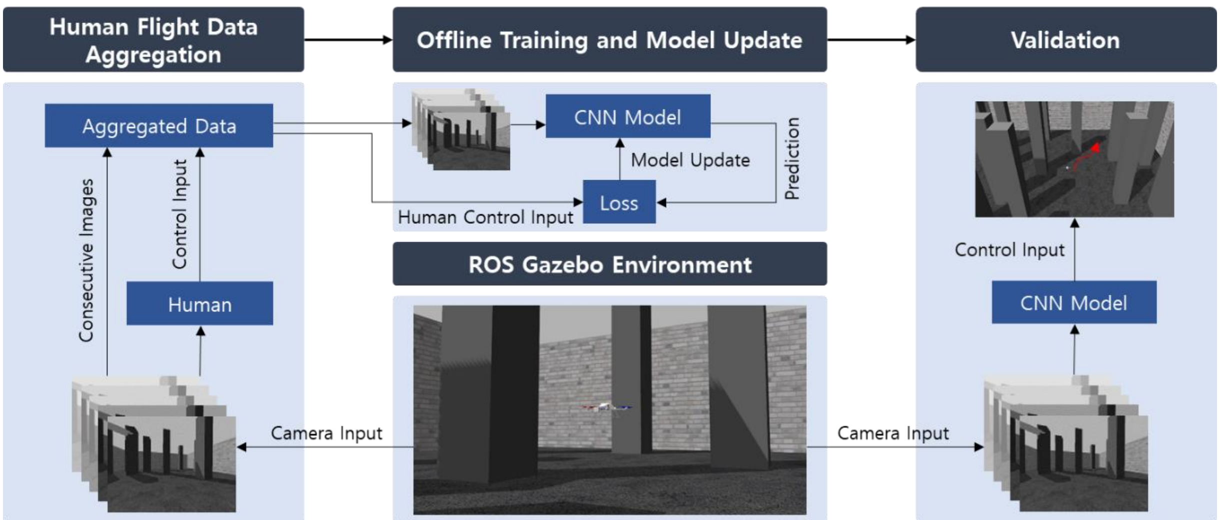


Figure 1 – Proposed imitation learning framework

2 BACKGROUND ON NEURAL NETWORKS

The advent of deep learning technologies has brought about substantial changes throughout various fields of technology. Based on multiple layers of artificial neural networks, deep learning is well known for its ability to approximate complex nonlinear functions, performing complicated classification or regression tasks.

Unlike fully connected layers, Convolutional Neural Networks (CNN) [13] share weights by using a convolution filter among its layers. This enables the neural network to not only reduce the number of weights, but also automatically extract local features from the input. Such characteristics grant CNNs a great advantage over other fully connected neural networks especially when it comes to image classification.

For conventional CNNs, a convolution operation is conducted for each layer with a 2D kernel granting CNNs considerable capabilities regarding image processing. At the same time, CNNs possess the same limitations that other feedforward neural networks have; feedforward neural networks are unable to consider sequential data. Although attempts to combine CNNs with RNNs (Recurrent Neural Networks) [14] [15], have proved to be efficient tasks such as in video classification and scene labelling, 3D-CNNs have also been renowned for its capability to extract spatiotemporal features from a given sequence of data. 3D-CNNs utilize a 3D kernel for the convolution operation, overcoming the previous limitations of 2D-CNNs. As obstacle avoidance is considered a sequential task, 3D-CNNs will be utilized in this study.

3 APPROACH

3.1 Imitation Learning Framework

For a drone to avoid obstacles during autonomous navigation, we utilize deep learning techniques to approximate the function between the visual inputs and controls. We first start by aggregating image and control input data from human flight demonstrations. During the demonstration, a human expert steers a simulated drone within the Gazebo environment. The accumulated data is then processed so that it can be used for the training of the neural network. Using this training data, the training process of the neural network is done offline. After the training process is finished, the neural network model is capable of imitating human expert decisions. For real-time applicability, a Gazebo-Python communication node is established using ROS (Robot Operating System). Here, the Python node receives images from the Gazebo simulator, where each image is fed through the neural network producing control input predictions for what a human expert may have chosen. Hence the drone is able to mimic expert behaviours, and consequently avoid obstacles. The overall framework is shown in Figure 1.

3.2 Training Data Acquisition and Processing

The training data for the neural network was obtained from the Gazebo simulator, using ROS. A human expert demonstration of obstacle avoidance within the simulated environment was recorded, i.e., control inputs from the human expert flight and the corresponding image data was aggregated into the training dataset. The sampling rate for the images was 15 frames per second, which is identical for the sampling rate of the control inputs. For each control input data, a value of 0, 1, or 2 is saved to indicate whether the human expert has commanded the drone to steer left, right, or move straight, respectively. The drone increases or decreases its heading angle upon left or right control inputs, and incrementally returns its heading angle to 0 when a ‘go straight’ command is given. A total of two hours of flight data from the expert was collected. Considering the fact that the environment did not exhibit much variance with respect to the colour and that a grayscale image contains adequate information for the identification of obstacles, each image from the training dataset was converted into a grayscale image. After the whole dataset is processed so that it can be used for training, we separate the dataset into a training and test dataset with a ratio of 7:3. We use the training dataset to train the neural network model, and the test dataset to verify the performance of the neural network. An example of the training environment and training image is shown in Figure 2.

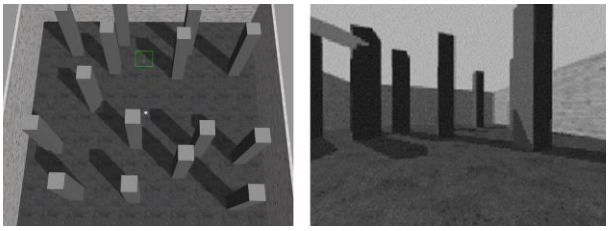


Figure 2 – Training environment (Left) and camera data (Right).

3.3 Neural Network Architecture and Training

As we are dealing with sequences of image data, we used a 3D-CNN architecture for the experiment. A total of 4 convolutional layers and 2 max pooling layers were used, and a fully-connected hidden layer was added at the end of the network. A Softmax output layer is added at the end of the network. The illustration of the CNN structure utilized in this study is shown below in Figure 3. For each layer, the first number indicates the step size of the images used for each batch, the next two numbers indicate the size of the input for each layer, and the last number indicates the number of channels or filters used. The size of the convolution filter was 3x3x3 and a 2x2x2 kernel was used for the pooling layer.

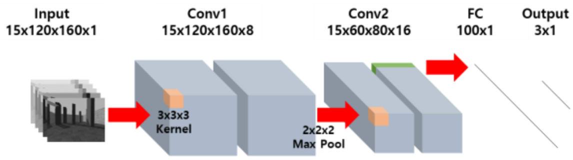


Figure 3 – 3D-CNN architecture for imitation learning

The training process of the 3D-CNN is done offline using the training data above. The 3D-CNN was trained with an epoch of 6,000, and the learning rate was 0.001. We used Adam Optimizer as the optimizer and truncated normal initializer for the initializer.

4 SIMULATION RESULTS

The imitation loss is as shown in Figure 4. The blue lines indicate the training loss whereas the red lines indicate the test loss. The final training loss was 0.006. The test loss was obtained by feedforwarding images from the test dataset, which were not used during the training process. The training accuracy was 96% and the test accuracy was 81%. Despite the minor discrepancy between the prediction values and labels for the test dataset, we can imply that the 3D-CNN is capable of mimicking human decisions given a set of images.

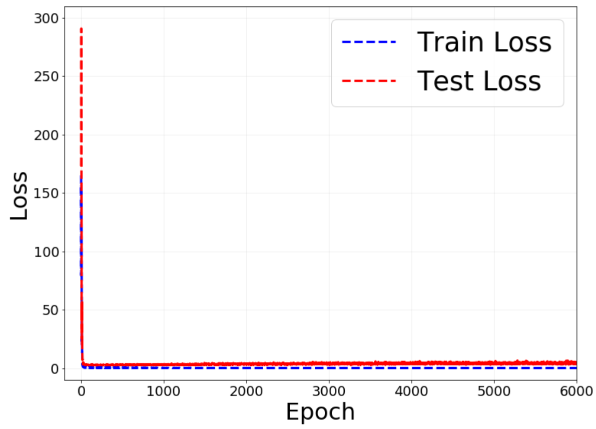


Figure 4 – Training and Test Loss

We applied the 3D-CNN to the drone to verify the effectiveness of the trained model regarding obstacle avoidance, implementing the real-time Gazebo-Python communication node using ROS.

Three unknown environments with randomly placed obstacles were used for the validation of the obstacle avoidance capability. Unlike the rectangle map used for the aggregation of the training data, the test map was designed as a long rectangular figure so that the drone could be tested if it is capable of crossing the entire map without crashing into any obstacles. In addition, the obstacles were arranged in a manner that the drone has not been able to see during the training phase to test the robustness of the obstacle avoidance capability. This demonstrates the fact that the 3D-CNN has learned a general rule for obstacle avoidance using a monocular camera, instead of learning how to navigate within the training environment only. Figure 5 shows an example of one of the maps used for the validation. The trajectory of the drone is shown in Figure 6. The trajectories show that the drone was able to successfully cross the entire maps in all three test maps, suggesting that the 3D-CNN has learned how to mimic a human expert and avoid obstacles during navigation.

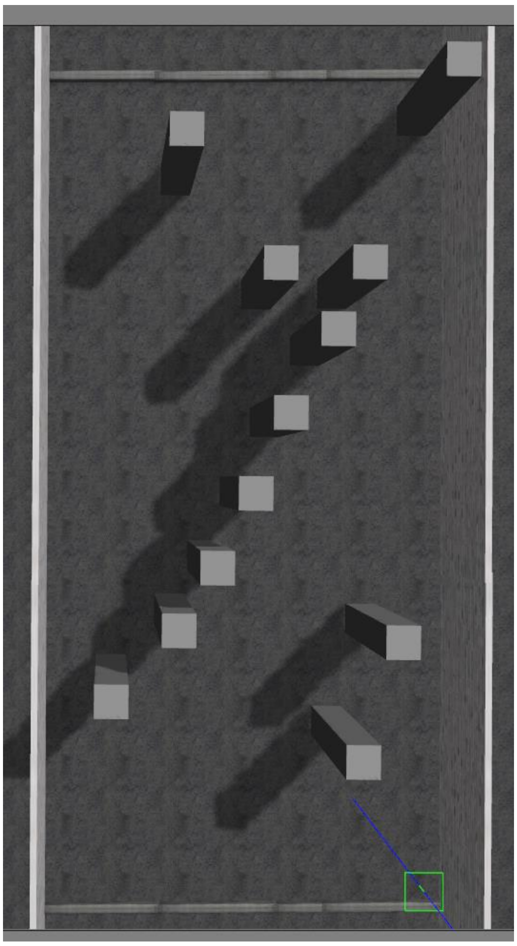


Figure 5 – Test environment example

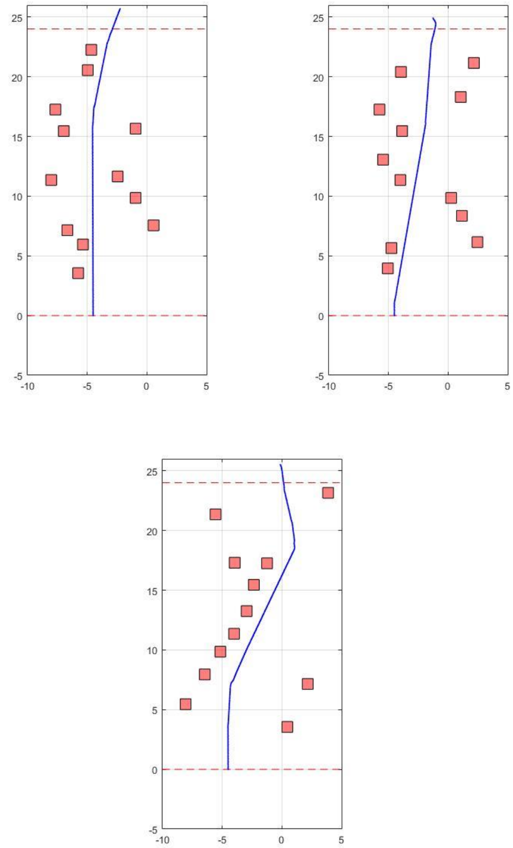


Figure 6 – Drone trajectories within the three test environments

5 CONCLUSION

An imitation learning framework using 3D-CNNs for obstacle avoidance has been investigated in this paper. Through our experiments with a drone in a simulated environment, the 3D-CNN model exhibited human-like behaviour, and at the same time we were able to achieve acceptable results for obstacle avoidance without the utilization of complicated neural network structures, proving the feasibility of applying imitation learning to drones to conduct a certain task.

Considering the fact that the imitation loss dramatically decreased after the first few epochs, the obstacle avoidance task can be considered a rather simple task to train, and thus the applicability to more complicated environments or more sophisticated tasks are yet to be explored.

ACKNOWLEDGEMENTS

This work was supported by the technology innovation project (S2439592) of Small and Medium Business Administration.

REFERENCES

[1] Chu, K., Lee, M., & Sunwoo, M. (2012). Local path planning for off-road autonomous driving with avoidance of static obstacles. IEEE Transactions on Intelligent Transportation Systems, 13(4), 1599-1616.

[2] Borenstein, J., & Koren, Y. (1988). Obstacle avoidance with ultrasonic sensors. IEEE Journal on Robotics and Automation, 4(2), 213-218.

[3] Borenstein, J., & Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. IEEE transactions on robotics and automation, 7(3), 278-288.

[4] Souhila, K., & Karim, A. (2007). Optical flow based robot obstacle avoidance. International Journal of Advanced Robotic Systems, 4(1), 2.

[5] Song, K. T., & Huang, J. H. (2001). Fast optical flow estimation and its application to real-time obstacle avoidance. In Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on (Vol. 3, pp. 2891-2896). IEEE.

[6] Kahn, G., Zhang, T., Levine, S., & Abbeel, P. (2017, May). Plato: Policy learning using adaptive trajectory optimization. In Robotics and Automation (ICRA), 2017 IEEE International Conference on (pp. 3342-3349). IEEE.

[7] Pomerleau, D. A. (1989). Alvinn: An autonomous land vehicle in a neural network. In Advances in neural information processing systems (pp. 305-313).

[8] Sammut, C., Hurst, S., Kedzier, D., & Michie, D. (1992). Learning to fly. In Machine Learning Proceedings 1992 (pp. 385-393).

[9] Ross, S., Gordon, G., & Bagnell, D. (2011, June). A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the fourteenth international conference on artificial intelligence and statistics (pp. 627-635).

[10] Ross, S., Melik-Barkhudarov, N., Shankar, K. S., Wendel, A., Dey, D., Bagnell, J. A., & Hebert, M. (2013, May). Learning monocular reactive uav control in cluttered natural environments. In 2013 IEEE international conference on robotics and automation (pp. 1765-1772). IEEE.

[11] Kim, D. K., & Chen, T. (2015). Deep neural network for real-time autonomous indoor navigation. arXiv preprint arXiv:1511.04668.

[12] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4489-4497).

[13] LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010, May). Convolutional networks and applications in vision. In ISCAS (Vol. 2010, pp. 253-256).

[14] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2625-2634).

[15] Pinheiro, P. H., & Collobert, R. (2014). Recurrent convolutional neural networks for scene labeling. In 31st International Conference on Machine Learning (ICML) (No. EPFL-CONF-199822).

Computer Vision Based Solutions for MAV Target Detection and Flight Control

M. F. S. Neto[1,2*], G. S. Eduardo[1,2*], E. C. Silva[1,2*] and W. Caarls[1*]
[1]Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro, PUC-Rio, Rio de Janeiro, Brazil
[2]AeroRio, Pontifical Catholic University of Rio de Janeiro, PUC-Rio, Rio de Janeiro, Brazil

Abstract

Computer vision applied to localisation and target detection has been a field of study in the literature for some years. Conventionally, MAVs (Micro Aerial Vehicles) used to rely solely on distance sensors and ran on simple and modest embedded devices. The huge increase of computational power made possible the use of more complex computer vision algorithms for real-time embedded applications. This paper focuses on evaluating different types of computationally demanding algorithms, such as accumulator-based image transforms (Hough, Radon), mathematical morphology and Monte Carlo approaches, to process data fed by a single camera in order to aid a MAV to navigate through an obstacle course.

1 Introduction

In recent years, there has been a marked increase in research related to multicopters and UAVs (Unmanned Aerial Vehicles). Recent market research showed that the global market revenue of drones was worth US$ 6.0 billion in 2017 and is expected to grow up to US$ 11.2 billion by 2020 [1]. The development of UAVs has been driven by recent advances in computational technology, software development, lightweight materials, global navigation systems, advanced data links, sophisticated sensors and component miniaturisation.

Some UAVs may navigate autonomously by continuously monitoring data from IMUs and a GPS. However, in order to perform complex tasks in confined areas, small autonomous drones will need more complex levels of autonomous control and extra sensors in order to identify features of its surroundings and perform safe and stable trajectories. Computer vision is an often used method of sensing for small UAVs due to its reduced mass and energy consumption compared to other methods, such as LIDARs and sonars [2].

This work aims to develop computer vision-oriented control and decision-making algorithms in order to allow a MAV (Micro Air Vehicle) to perform an obstacle course completely autonomously. More specifically, the hoops element of the IMAV2018 course, that consists in making the MAV fly through a sequence of five hoops, with ellipsoidal geometries, with five different sizes. The smaller the hoops, the higher the score, but it increases the complexity of the task as well.

The objective is to properly identify a hoop and infer its position referential to the MAV, using image processing techniques. Studies will be carried out on classical methods of image processing for object detection [3] while evaluating different algorithms by their accuracy, processing time, noise and deviation, to verify their performance in the accomplishment of the tasks of interest.

2 Methodology

The developed MAV is based on the Emlid Navio2 flight controller paired with a Raspberry Pi 3 microcomputer. The system will receive environment inputs via a single Raspberry Pi Camera, pointed forward.

For the hoop detection (a hollow ellipse), which in the present work is considered to be almost a circle, the goal is to estimate the coordinates of the centre of the shape, x_o and y_o, and the radius, ρ. In order to achieve that, a mix of different circle detection approaches (Accumulator-based and Monte Carlo) with different methods of image pre-processing (mathematical morphology, clustering and fuzzy segmentation) were used. In total, 7 different combinations were evaluated, they are: Random Sampling Consensus (RANSAC) using a binary image; Randomised Circle Detection (RCD) using a binary image; RCD using a binary image obtained by clustering; Radon Transform over a fuzzy-segmented image; Hough Transform using a binary image obtained by clustering; Hough Transform using a fuzzy-segmented mask; Hough Transform using a greyscale image obtained from the Lab colour space. All the methods were implemented using the Python language and the OpenCV library.

2.1 Accumulator-based approaches

Accumulator-based approaches use a ”voting array” with N dimensions, each corresponding to a parameter of the shape to be detected. In case of a hoop, the three necessary parameters results in a 3-dimensional array. The two most common methods for accumulating the array are the Hough transform and the Radon transform.

The conventional way of implementing the Hough method involves determining, for every segmented pixel in

*Email address(es): manoelcfeliciano@gmail.com, guiseduardo@gmail.com, edusilva@ele.puc-rio.br, wouter@ele.puc-rio.br

a binary image, all possible circles that might contain said pixel, and then mapping the parameters of these circles to the accumulator array. Each element of the accumulator array represents a number of votes for the possible circles present in the image. The presence of the local maximums indicate strong evidence of circles with parameters described by their coordinates in the accumulator array [4]. Building a 3D accumulator array is very computationally demanding. In case of large images, the controller performing this kind of algorithm could experience memory issues, since the large amount of possible parameters generates an even larger accumulator array [4]. Because of this, the OpenCV implementation of the Hough transform for circles uses a technique called the Hough Gradient Method that uses just a 2D accumulator. This method receives, as function parameters, a radius threshold, limiting the radius of found circles to a minimum of 10 pixels and maximum of 250 pixels, high threshold for the Canny edge detection of 50, the low being twice smaller, and the accumulator threshold for circle centres of 35. These parameters were empirically chosen by experimenting with a variety of images and footage.

As for the Radon method, the density of each element of the accumulator array leads to the match between the image and a template generated using the parameters given by the coordinates of the given element [5]. It is calculated by the inner product between a segmented image emphasising the hoop (e.g. using a fuzzy membership function) and the projection of the template into a blank image.

2.2 *Monte Carlo approaches*

Monte Carlo approaches form a class of algorithms that relies on the repeated random sampling of the process inputs. They come as an alternative when numerical or analytic strategies are not practical or possible for the solution. Furthermore, they benefit from the fact that they do not need an accumulator. There are two common methods for detecting circles which use this kind of approach: the Random Sample Consensus (RANSAC) and the Randomized Circle Detection (RCD) algorithms.

The RANSAC algorithm proposed by Fischler and Bolles [6] is a classic implementation of this approach in the computer vision field. Its working principle is based on the robust estimation of a number of parameters from a model using a random number of hypothesis. The main difference from other common robust parameter estimation algorithms is the use of the smallest number of observations possible to obtain the initial solution of the problem. This is achieved by first solving for the model using three random samples each time and then verifying the degree of trust of the estimations based on the inliers that follows a predefined threshold. The output of the algorithm is given by the estimation with the highest degree of trust in a voting procedure. Therefore, the main advantages of the algorithm lie not only in its robustness to the presence of outliers but also in its efficiency. It is important to note that, for the RANSAC method in this work, the edges of the hoop are not detected using Canny edge detection, but using the external contour retrieval OpenCV function.

Following a different strategy, the RCD algorithm proposed by Teh-Chuan Chen and Kuo-Liang Chung is based on a voting procedure in the parameter space. It works by first selecting 4 random edge pixels from the image using a distance criterion to determine the existence of a possible circle. Further, it verifies the circle candidate by using an evidence collection process such as the number of edges pixels that lies inside of it [7]. This algorithm uses a number of thresholds to achieve the previous strategy with the following empirically determined values for the test environment. The distance threshold, limits the distance between a circle candidate and each edge pixel to 40 pixels. The ratio threshold, limits the ratio between each edge pixel and the number of pixels in the boundary of the circle candidate to 0.5. The minimum distance between two edge pixels of a possible circle is 10 pixels. The circle detection task is stopped if there are less than 10 edge pixels in the set. Finally, the maximum number of attempts in detecting a circle is set to 10.

2.3 *Image pre-processing*

As described in sections 2.1 and 2.2, each method needs to work with a segmented image, that is, an image that emphasises the hoop and/or its features by combining different colour space channels of the capture, so computer vision algorithms such as the Hough Transform can easily detect the correct edges in the image, as the present noise is mitigated. The standard colour space used by cameras is the RGB (Red-Green-Blue). In this work, however, the segmentations are obtained using the HSV (Hue-Saturation-Value) and Lab (Lightness-green/red-blue/yellow) colour spaces, because these setups enhance the contrast between the hoop and the background.

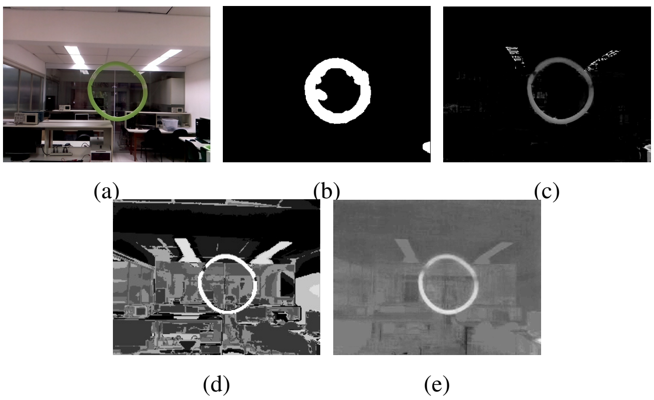


Figure 1: Different methods of image pre-processing. (a) Original frame. (b) Binary segmentation. (c) Fuzzy segmentation. (d) KMeans mask. (e) Lab 'a' channel.

The simplest form of segmentation is the generation of a binary image by verifying the pertinence of each pixel to a previously-set threshold. In the case of the HSV channels, applying a threshold to the 'Hue' channel, for example, will emphasise the pixels belonging to a certain range of colours. Figure 1b is an example of a binary segmentation. This kind of image can be further improved by applying morphological operations, such as erosion, to remove noise, and dilation, to close gaps in the image. For the RANSAC algorithm, the image is generated by applying a threshold to the 'Hue', Saturation and Value channels, respectively, at the intervals (30;60), (32;255) and (50;255) and, then, eroded once and dilated 4 times in order to remove noise and close gaps. On the other hand, for the RCD algorithm, the image is generated by applying a threshold only to the 'Hue' channel, at the interval (30;80) and, then, dilated 3 times. These ranges were empirically established by experimenting with a variety of images and footage.

The Fuzzy segmentation works likewise, but instead of generating a binary image, it generates a greyscale image where the value of each pixel represents its membership grade to the specified group of parameters (in the HSV image, the parameters would be colour, saturation and brightness). The obtained image is, then, used as a mask over the original greyscale image, resulting in Figure 1c. In this work, the fuzzy membership function used is just a trapezoidal function applied to the 'Hue' channel, with parameters P = (20;40;60;80).

The KMeans segmentation uses the Scikit-learn clustering function to identify clusters of pixels with the same colours in the HSV space. Then, this information is used to generate a mask that better emphasises the hoop and hides unwanted portions of the image. Figure 1d shows a binary image used as a mask, obtained by the KMeans method.

3 Tests and Results

3.1 *Preliminary observations*

During the development phase, each hoop detection method was tested against a set of pictures containing hoops similar to the ones used in the obstacle course. This is to ensure that every method would display reasonable levels of detection rate, accuracy and processing time.

One of the most important parameters for using computer vision in drones is the processing time of each frame. Optimisations such as subsampling the frames down to 160 px width, maintaining the aspect ratio, had to be made in order to improve the real time performance. However, this was not enough for the Radon method which was able to correctly detect the hoop, but needed 37 minutes and 46 seconds to process a single frame in the Raspberry Pi, an impractical amount of time for live applications. The result for this particular algorithm was expected since it is based on a 3D accumulator and does not have any optimisation, other than image subsampling, implemented. Thus, this method was not included in the next phase of testing. It should be noted that better results are expected with a GPU due to the parallelizable nature of this method.

3.2 *Live testing*

This phase of testing consisted in comparing the performance of each algorithm using the MAV's hardware for varying positions in relation to the hoop, as well as different viewing angles. The MAV was kept at the same position on a still surface at all times with its motors turned off.

In each test, the Pi Camera captured a set of 100 frames after which the Raspberry Pi proceeded to process the frames using the methods chosen, finding the parameters of the hoop in pixels, proportional to the camera resolution. The parameters were then combined with known information, such as the actual average radius of the hoop (the hoop used is ellipse-shaped, with minimum diameter of 780 mm and maximum diameter of 860 mm, an average of 820 mm), camera resolution and field of view (for the Pi Camera, it is 62.2° horizontally and 48.8° vertically), in order to estimate the relative position between the MAV and the hoop. Thereafter, the methods were evaluated by the root mean square error (RMSE), standard deviation and confidence interval (95%, normal distribution) of the measured longitudinal distance to the hoop and the euclidean distance (Y and Z axes) to the centre of alignment, processing time and the ratio of valid detections.

First, the hoop was positioned at the minimum distance of detection from the MAV, 140 cm. Then, in steps of 30 cm, the hoop was brought farther from the MAV until the distance of 410 cm. In this test, not only the perceived radius of the hoop was smaller, but the perceived thickness of the edges was also thinner, affecting the image segmentation. Furthermore, the changes in luminosity in different parts of the room where the tests were conducted might interfere with the hoop detection, requiring each algorithm and image segmentation method to be robust against these changes.

Algorithm	Pre-processing (ms)	Detection (ms)
Hough Fuzzy	16.90 ± 1.12	6.38 ± 5.13
Hough KMeans	438.78 ± 166.69	62.43 ± 14.48
Hough Lab	9.64 ± 0.53	4.28 ± 0.70
RANSAC	44.25 ± 2.60	23.22 ± 6.71
RCD	44.25 ± 2.11	50.71 ± 13.59
RCD KMeans	441.11 ± 177.52	1334.39 ± 564.57

Table 1: Average times (± standard deviations) required for each algorithm to process one frame and detect the hoop.

Table 1 shows the average processing time plus standard deviation required for each method to pre-process an image and detect the hoop over all the distances tested. All the methods kept consistency even when the rates of detection seen in Figure 2a dropped to very low values. The processing times presented by Hough Fuzzy and Hough Lab methods were fast enough for high-framerate applications. RANSAC and RCD reached higher processing times, but still good enough for

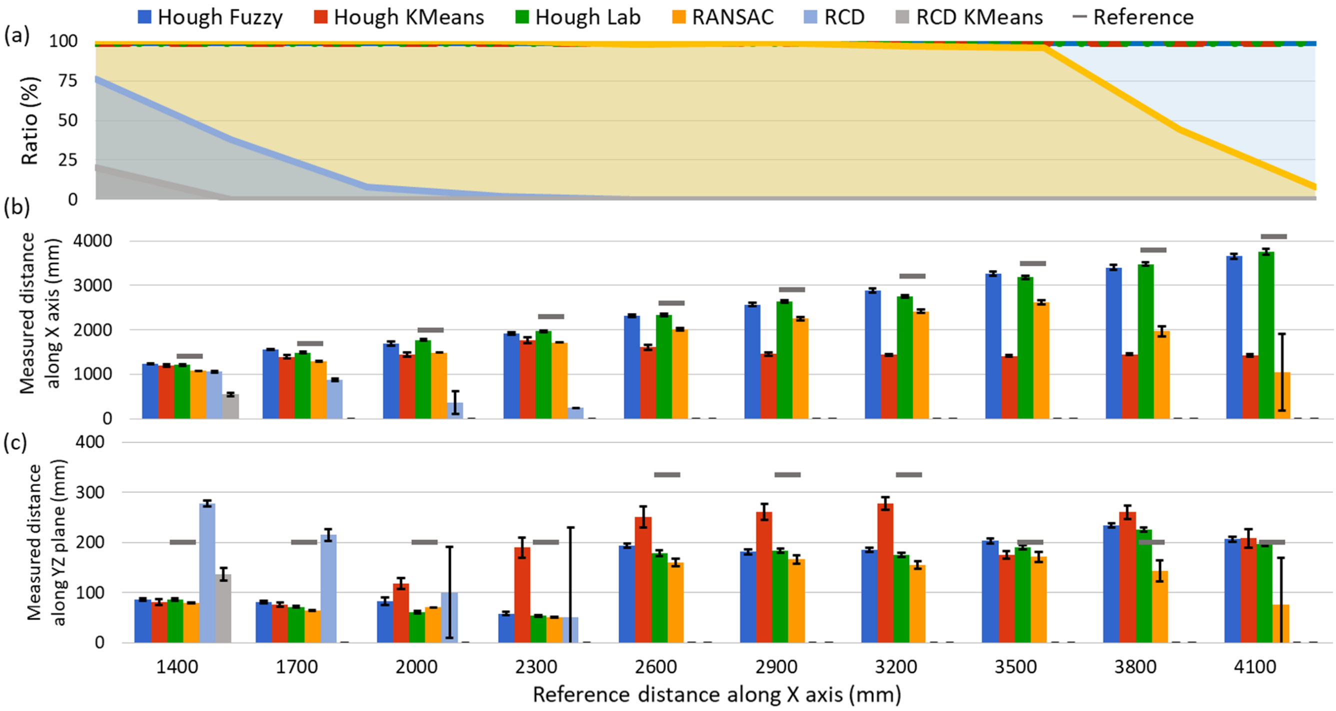


Figure 2: Algorithm performance for different distances between 1400 mm and 4100 mm. (a) Ratio of valid detected frames. Notice how all the Hough methods achieved 100% detection. (b) Estimated distance along longitudinal axis for each algorithm, with reference distances and confidence interval of 95%. (c) Estimated distance over the frontal plane (YZ axes) for each algorithm, with reference distances and confidence interval of 95%. The reference distance along the YZ plane is (0;-200) mm for all reference distances along the X axis, except between 2600 mm and 3200 mm, where it was set to (-300;-150) mm due to placement difficulties in the testing room.

the desired application. The processing time is noticeably affected by the image pre-processing method chosen, as the methods that use the KMeans clustering took longer on average to process the segmented frame.

Figure 2a shows how the detection rates of the hoop drops as the distance from the camera increases. All Hough methods achieved 100% detection rate at all distances, while the detection rate of the other methods tends to decrease from a certain distance. This happens because the hoop becomes very thin at increased distances, becoming faded after morphological operations, impairing detection by the Monte Carlo-based methods. The Hough-based methods are not affected because they rely on Canny edge detection, identifying the presence of the hoop as long as it contrasts with the background.

Figure 2b displays the evolution of the average distance estimation and confidence interval along the longitudinal axis (X axis), calculated based on the ratio between the hoop size in the capture, in pixels, and the actual hoop size, in millimetres. Figure 2c displays the average estimation and confidence interval of the euclidean distance over the YZ plane (vertical and lateral axes) between the hoop and the centre of the capture.

It can be said that the Hough Fuzzy, Hough Lab and RANSAC methods were the best performers in this test by demonstrating high detection rate, low processing time, the least errors compared to other methods and little dispersion of results. It can also be concluded that the perceived radius of the hoop is greater than its actual radius, leading the algorithms to estimate shorter distances to its centre. This effect may be due to the eccentricity of the hoop, causing the detected radius to be its semi-major axis instead of its mean radius, preset in software.

The other methods perform worse due to various reasons. Besides the long processing time, the Hough KMeans method couldn't correctly identify the hoop, as evidenced by the almost constant distance estimate regardless of the actual distance. Furthermore, both RCD methods could only properly detect the hoop within close proximity, making them unreliable for longer distances.

Another round of tests consists on evaluating the performance of each algorithm in detecting a rotated hoop. The motivation for this kind of test is that the MAV might not always be perfectly aligned with the hoop, requiring each algorithm to detect it nonetheless. In addition, each algorithm will perceive the hoop as an ellipse with increasing eccentricity as the

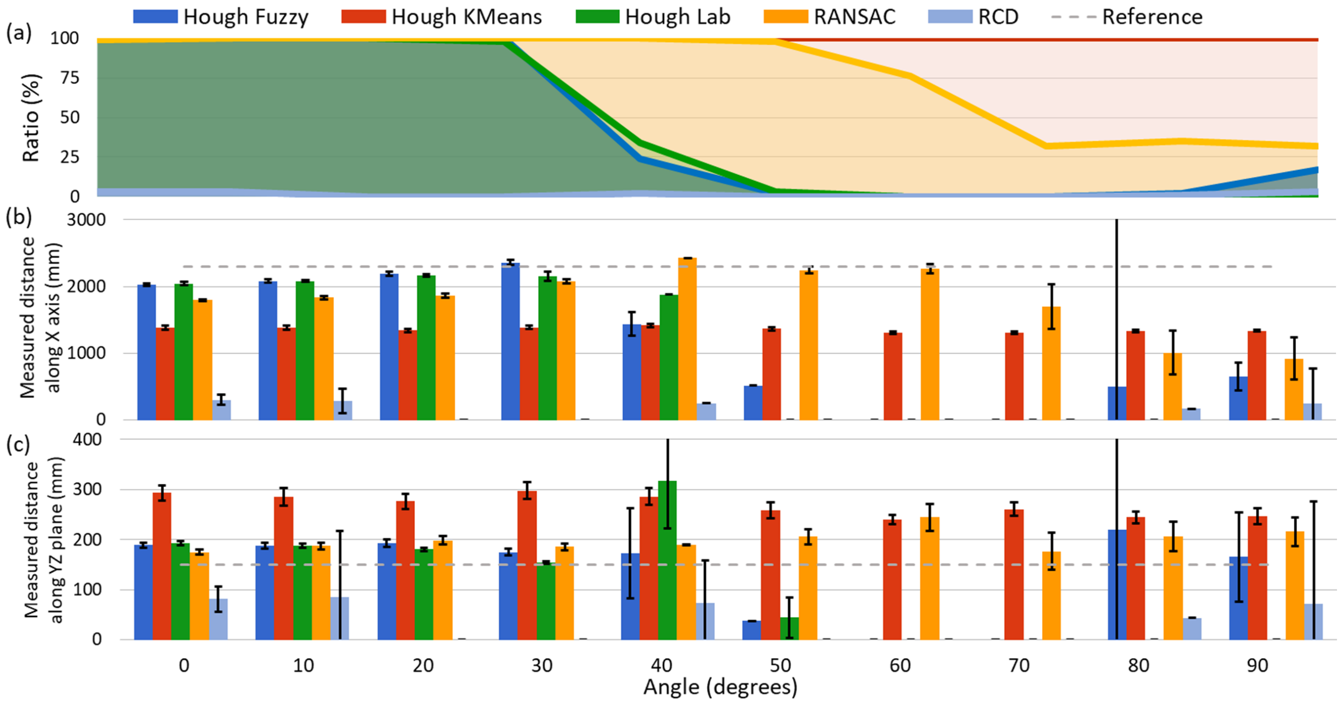


Figure 3: Algorithm performance for different hoop rotation angles at a distance of 2300 mm. (a) Ratio of valid detected frames. At this distance, both RCD methods had very low and 0 detection rates. (b) Estimated distance along longitudinal axis for each algorithm, with reference distances and confidence interval of 95%. (c) Estimated distance over the frontal plane (YZ axes) for each algorithm, with reference distance ((0;-150) mm for every test) and confidence interval of 95%. The three algorithms with best results (Hough Fuzzy, Hough Lab and RANSAC) kept consistency in measurements with good detection ratios.

angle of rotation increases. The routine for this test is similar to the previous one: starting from 0° rotation (with the MAV completely facing the hoop head on), in 10 steps of 10°, up to 90° (hoop sideways in relation to the MAV), each algorithm evaluates a set of 100 frames per step.

This test was performed by placing the hoop at 2300 mm from the camera. This distance was chosen because the hoop occupies a reasonable area of the frame. However, both RCD methods underperform, as shown in Figure 2b. In spite of that, both of them were also subjected to these new tests. The RCD KMeans results were omitted in Figure 3, because its valid detection frames ratio remained null for all analysed cases.

Figure 3a shows the detection rate of valid frames as the angle of the hoop increases. As expected, detection rates drop past certain angles, except for the Hough KMeans method that, as observed in previous tests, might be detecting noise instead of the actual hoop.

Figures 3b and 3c show the distance estimations along, respectively, the longitudinal axis and the YZ plane, as the rotation of the hoop varies. As expected, measurements at angles where detection rates are low are mostly noise, as evidenced by the greater dispersion of results past these angles.

An interesting phenomenon in the measurements was a steady increase in the distance estimation by the Hough Fuzzy, Hough Lab and RANSAC methods while keeping low dispersion. As previously said, the estimated distance is calculated from the ratio between the measured radius from the capture, in pixels, and the actual radius of the hoop, in millimetres. This leads to the conclusion that the estimated radius found by these algorithms becomes smaller with the increase of eccentricity of the hoop seen from the MAV's point of view.

As a final test, each detection method is evaluated on its ability to detect the hoop while it is partially outside the camera field of view. This test is performed at the constant distance mark of 2300 mm by the longitudinal axis and increasing distances by the lateral axis, so the hoop can be partially visible by 25%, 50% and 75% of its total area. The 100% and 0% visibility tests are also included for comparisons.

Figure 4a shows how the detection rate drops as the hoop moves away from the field of view. The obtained results indicate that the Hough Fuzzy, Hough Lab and RANSAC methods still can identify the hoop even in situations where it is not fully contained in the frame.

Figures 4b and 4c show how the visible hoop area affect the distances estimated by the algorithms. The results confirm that Hough Fuzzy, Hough Lab and RANSAC algorithms can

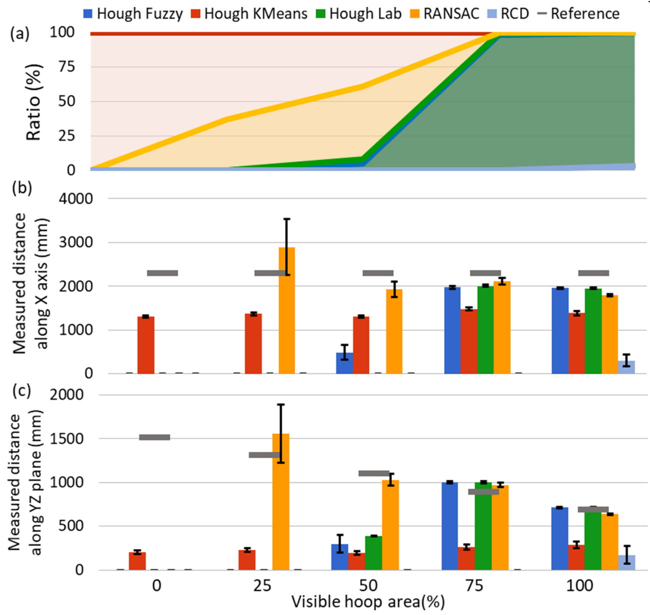


Figure 4: Algorithm performance as the hoop moves outside the camera field of view. (a) Ratio of detected frames. (b) Estimated distance along the longitudinal axis for each algorithm, with reference distance and confidence interval of 95%. (c) Estimated distance over the frontal plane (YZ axes) for each algorithm, with confidence interval of 95% and reference distances of (-1510;-150) mm, with the hoop outside the frame, and the distance along the Y axis increasing in steps of 210 mm until (-670;-150) mm, when the hoop is fully into view.

correctly identifying the hoop position and radius, in spite of the partial concealment of the hoop. For visible hoop areas as low as 75%, these three methods measured similar distances during valid results, along the longitudinal axis and the YZ plane. However, regarding this aspect, it is noted that the RANSAC method outperforms the others, since it returned adequate measurements for visible hoop areas as low as 50%, while others start to underperform earlier.

4 Conclusion

There are many different algorithms used for implementing computer vision for different applications, each one with their own advantages and disadvantages. This work evaluated the performance of different combinations between 4 image segmentation methods (binary, fuzzy, Lab grayscale and clustering) and 4 circle detection algorithms (Hough transform, Radon transform, RANSAC and RCD), focusing in identifying and measuring a hoop in order to aid a MAV to traverse it. The best suited algorithms for the task were selected by comparing their better performance and reliability. The Hough Fuzzy, Hough Lab and RANSAC algorithms presented shorter processing time, higher detection rates and better accuracy.

These 3 methods performed well, however they have some distinctions that are relevant to the objective of traversing the 5 hoops of the obstacle course. The RANSAC method works really well for adverse situations, such as detecting a hoop with high eccentricity (or rotated more than 40°) or a hoop that is not fully viewed by the camera (regular detection rate for a visible area as low as 50%). However, the MAV might not have to deal with these kind of situations, making the Hough approaches preferable, because of their faster processing times and better accuracy, especially the Hough Lab method, that presented the best performance in the tests conducted.

In future works, new tests will be conducted with a flying MAV in order to evaluate different algorithms, considering the effects of vibration and motion.

Furthermore, the performance of other methods and their combinations with pre-processing algorithms will be evaluated, including methods based in deep learning that, with the appropriate hardware support, have the potential to be more accurate.

Acknowledgements

We would like to thank the Brazilian funding agencies CNPq, FAPERJ and CAPES for continued support and supplied resources.

References

[1] C. Stamford. Gartner says almost 3 million personal and commercial drones will be shipped in 2017, 2017. https://www.gartner.com/newsroom/id/3602317 (Accessed on 05/05/2018).

[2] D. Floreano and R.J. Wood. Science, technology and the future of small autonomous drones. *Nature International Journal of Science*, 521:460–466, 2015.

[3] P. Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Publishing Company, New York, NY, 2013.

[4] H.K. Yuen, J. Princen, J. Illingworth, and J. Kittler. Comparative study of hough transform methods for circle finding. *Image and Vision Computing*, 8(1):71–77, 1990.

[5] M. van Ginkel, C.L. Luengo Hendriks, and L.J. van Vilet. A short introduction to the radon and hough transforms and how they relate to each other, 2004.

[6] 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.

[7] T.C. Chen and K.L. Chung. An efficient randomized algorithm for detecting circles. *Computer Vision and Image Understanding*, 83:172–191, 2001.

Autonomous landing algorithm using a sun position predicting model for extended use of solar powered UAVs

B.P. Duisterhof and G.C.H.E. de Croon

ABSTRACT

In the field of robotics, a major challenge is extending the flight range of micro aerial vehicles. One way to extend the range is by charging batteries with solar arrays on the ground, while resting on intermediate landing positions. The solution we propose in this study differentiates itself from other solutions as it does not focus on improving UAV efficiency but rather on finding the most efficient landing position. In particular, an algorithm is developed to show the usefulness of the approach. This algorithm makes uses of the sonar sensor on board of the Parrot Bebop 1 drone in combination with an OptiTrack system to scan the environment for potential landing opportunities. After these measurements are discretized on a 2D grid, analysis is carried out with a sun position predicting model. Finally, a landing position is chosen within the scanned area and the drone will land accordingly.
Little is known on whether a solar powered charge on the ground could be effective in a limited period of time. We present a coarse analysis, showing that the DelftaCopter with solar arrays on its wings charges its batteries in 1.3 days with relatively cheap solar cells in Africa or Australia. Future work includes the use of computer vision instead of sonar as well as the ensurance of a safe landing position using vision.

1 INTRODUCTION

Future UAV operations will become more and more autonomous as human interaction is expensive and inefficient. This means there is a need for increasing UAV range in an autonomous fashion, limiting human interaction. Solar energy has a pivotal role in extending UAV range and has been used extensively to increase in-flight efficiency [1, 2, 3]. Other methods using piezoelectricity have also been considered in previous studies [4].
The main new concept introduced in this study is to use this technology with a new strategy. Very little is known about the use of solar arrays to charge UAVs while being at rest on the ground. This study was set out to develop an algorithm to charge a UAV at intermediate landing positions, using sonar and position data. With this the usefulness and feasibility is investigated.
A variety of applications exist, from delivering packages in remote areas (as the company 'Zipline' does [1]) to investigating large areas in nature [5, 6, 7]. UAVs capable of performing these type of tasks have been developed, but there might be cheaper options in the future using the new strategy. Mission time might be extended due to charging time, yet this could be compensated for by operating more UAVs in swarm formation.
In this study we will first evaluate previous work whereafter a sun model is selected for the purpose. Then, sonar and landing strategy are considered which is applied in the flight experiments section. Finally, sections on feasibility, conclusion and future work follow.

2 PREVIOUS WORK

Extending the range of UAVs has been subject of many studies in the field of robotics, mainly focusing on improving overall efficiency of aerial vehicles. Different energy harvesting methods have been considered in the past, such as the use of solar panels [1] or piezoelectricity (e.g. vibrations or rigid body movements)[4]. The AtlantikSolar from ETH [2] performed an uninterrupted flight of 81 hours using these cells. Also, vision based autonomous landing has been investigated before [8, 9].
To our knowledge, this is the first time that these two different topics are combined by selecting a landing spot not just based on safety, but also maximal exposure to the sun for fast recharging.

3 SUN MODEL

First the sun model will be discussed after which the scanning and environment analysis will be treated.
The sun position predicting model necessary for this application is elaborated on now. With this model, the UAV will finally be able to model the solar intensity on its surroundings and choose an appropriate landing position. To find the optimal landing position, it will calculate a 'solar-score', which is a measure for how illuminated a certain position is over a day of sun.

3.1 Sun model deliverables

It is of great use to describe the desired outcomes of the sun position predicting model, being: (i) Azimuth angle , a horizontal angle measured clockwise from a north base line

[1] `http://www.flyzipline.com/`, August 27 2018

or meridian, (ii) Elevation (altitude) angle, a vertical angle measured positive above the horizon. The physical meaning of these angles is depicted in Figure 1.

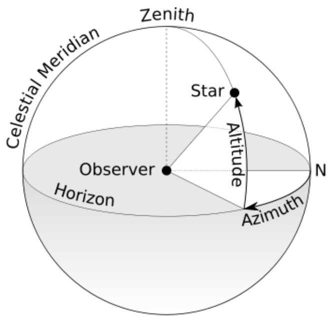


Figure 1: Elevation and Azimuth angles depicted on earth surface

Besides these deliverables, more requirements are present. Calculations should be computationally cheap enough to run on board a MAV and the calculated angles should be accurate enough to finally come up with the true optimal landing position.

3.2 Model decision

To finally compute the azimuth and elevation angle, a trade off between accuracy and computational effort is performed. The model from The Saudi Arabia University [10] is accurate and computationally cheap. It can predict both azimuth and elevation angle with an accepted maximum error of 3 degrees for azimuth angle and 1.4 degrees for elevation angle. To arrive at the deliverables, a number of steps is required which are listed now.

3.2.1 Local Standard Time Meridan (LSTM)

To start with, the LSTM can be calculated using Equation 1.

$$LSTM = 15 \cdot (\mid LT - GMT \mid) \tag{1}$$

Here LT is Local Time and and GMT Greenwich Mean Time. Both have to be inserted as a fraction in military time, s.t. 17:30 is inserted as 17.5.

3.2.2 Equation of Time (EoT)

The final goal of the Equation of Time (EoT) is to compute a standardized time : the Local Solar Time (LST). The equation is computed using curve fitting techniques, arriving at Equations 2 and 3.

$$EOT = 9.87 \cdot \sin(2B) - 7.52 \cdot \cos(B) - 1.5 \cdot sin(B) \tag{2}$$

$$B = \frac{(n-81) \cdot 360}{365} \tag{3}$$

Here n is the day number of the year, such that for January 1, n=1 and for December 31, n=365.

3.2.3 Time Correction Factor

Using the previously computed variables in Subsections 3.2.1 and 3.2.2, the total time correction factor can be computed according to Equation 4.

$$TC = 4 \cdot (Longitude - LSTM) + EOT \tag{4}$$

Here, longitude is included as UAV location is not necessarily synced with LSTM. In other words, LST is dependent on position and not only on the current time zone.

3.2.4 Local Solar Time

Finally, one standard time can be developed using Equation 5 : the Local Solar Time (LST).

$$LST = LT + \frac{TC}{60} \tag{5}$$

3.2.5 Solar Hour Angle (h)

The Solar Hour Angle will be advantageous in the final computation, seeing that it is defined as the angle w.r.t. the sun's orientation at its highest point. It is zero when the local solar time is 12 and 180 when the local solar time is 24.

$$h = 15^\circ \cdot (LST - 12) \tag{6}$$

3.2.6 Solar Declination Angle (δ)

Another angle necessary to finally compute the desired output is the solar declination angle. This angle is defined as the angle the earth's equator makes with the line joining the centers of earth and sun. This angle can be computed using Equation 7.

$$\delta = 23.45 \sin\left[\frac{(n-81)\,360}{365}\right] \tag{7}$$

This solar declination angle is useful for a range of astronomical applications as well as for this problem.

3.2.7 Solar Elevation angle (α_s)

Now, finally the first real output can be generated: the solar elevation angle. This is the angle the sun makes with the horizon and is often referred to as altitude. The solar elevation angle can be computed using Equation 8.

$$\alpha_s = \arcsin\left[\sin\delta\sin\phi + \cos\delta cos\phi\cos h\right] \tag{8}$$

Here ϕ is the latitude on earth where the model is used for.

3.2.8 Solar Azimuth Angle (γ_s)

The second desired output is the solar azimuth angle, which gives the orientation of the sun w.r.t. the north, in a clockwise fashion seen from above. It can be computed using Equation 9.

$$\gamma_s = \arccos\left[\frac{\sin\delta\cos\phi - \cos\delta\sin\phi\cos h}{\cos\alpha_s}\right] \tag{9}$$

This finally gives everything required to predict the sun's position. The curve fitting method assumes equality each year, which in reality is not entirely true. This is one of the sources which contributes to the final error in the model. Alternative methods make use of more dedicated curve fitting methods or even look-up tables. Future work includes a computation of the received energy per m^2 on the grid, including atmospheric conditions.

3.3 Solar Simulator Use

The equations discussed before have to be implemented in a computer program to test the outcome. This will be done in the C language to finally run on board of the Parrot Bebop 1 drone. A stand-alone program is set up first, with three main functions : (i) Manual Mode : compute elevation and azimuth angles for one specific point in time and space, (ii) Graph Mode: Write both angles for an entire day to a .txt file and (iii) Array Mode : Write an entire day into an array which can later be used elsewhere in the UAV. The use of the program is clarified in Figure 2.

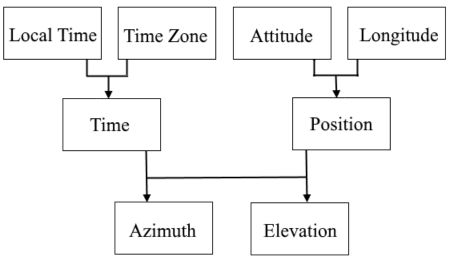


Figure 2: Program flowchart

The Graph Mode is built for verification purposes only, and will not be used in the UAV. Instead, only the array mode is used with location and time as inputs from the Bebop drone.

3.4 Verification

Now the outcomes of this program have to be verified and accepted. It has been shown earlier that the maximum elevation angle error is 1.4 degrees and the maximum azimuth angle error 3 degrees [10]. With this, two thought experiments are carried out: (i) a 2-D object (e.g. a plate), with error modelling of the present shadow over different elevation angles and (ii) a 3-D cylinder, with error modelling of sun position inside the cylinder over an entire day.

3.4.1 2D Case

The 2D case is relatively simple: the length of a shadow behind a wall is modeled with and without error. The three desired variables are: real shadow length, shadow length with error and the difference. General shadow length is defined by:

$$l_{shadow} = h_{obstacle} / \tan(\alpha_s) \tag{10}$$

Here l_{shadow} is the shadow length, $h_{obstacle}$ the obstacle height and α_s the elevation angle of the sun.
To compute the effect of model error, the maximum error will be added to the elevation angle. With this, the error can be calculated with the output as depicted in Figure 3 for a 2 m high object between 20 and 45 degrees elevation. This range has been chosen to demonstrate the behaviour of the error.

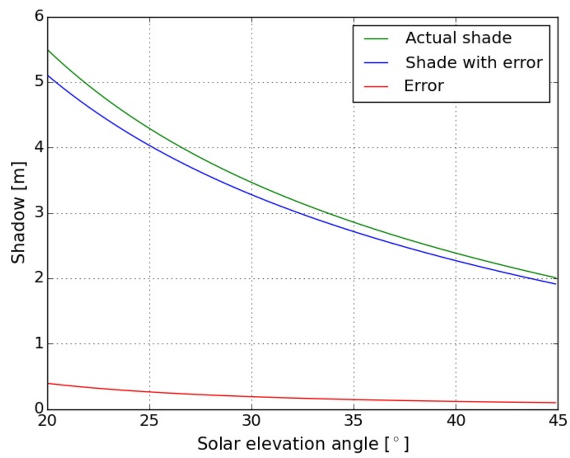


Figure 3: 2D shadow thought experiment for a 2m high object

To understand the behaviour, one can imagine a sunrise on a beautiful island. If the solar rays are coming in exactly horizontal, a palm tree will in theory create a shadow of infinite length. If then an offset of three degrees is added, a finite shadow arises with an infinite offset from the real shadow. In Figure 3 this tendency can be observed: for low elevation angles the offset increases. Anyhow, the error at one point in time is not enough to draw a conclusion for the final landing position. In subsection 3.4.2 the 3D case is analyzed for an entire day.

3.4.2 3D case

Fortunately, the problem is more complex. For the 3D case, a vertically placed cylinder is used in a thought experiment to test if the proposed model is indeed accurate enough. The goal is to obtain a solar energy map of the ground in the cylinder (which is placed vertical), showing where the most ideal landing position would be.

This thought experiment is carried out by computing the received sun over an entire day. The results are visible in Figure 4, where 100 means the maximum amount of sun is received at this point (100 %). X and Y are zero at the centre of the circle.

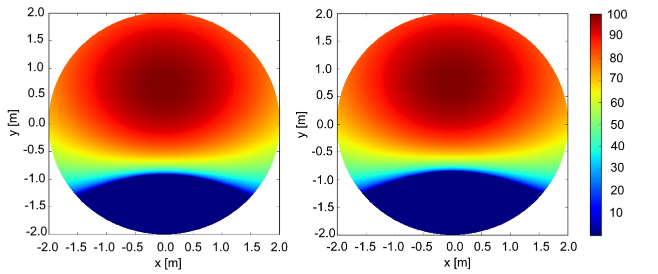


Figure 4: 3D cylinder thought experiment entire day. Left with error, right without error.

3.4.3 Model accuracy discussion

The results in Figures 3 and 4 have been computed numerically. The real important question should be : is the model good enough? Mainly focusing on Figure 4, it can seen that the landing position will not change significantly : the UAV will land in the middle of the dark red dot which is not moved significantly. Also, it is expected that the use of sonar will introduce far more significant errors than the model offset and will result in a relative coarse initial 2D mesh. With this, it is concluded that the sun-position predicting model is sufficiently accurate for the application.

3.5 Data acquisition

After the sun position predicting model has been established, a plan is made on how to obtain the required data. Two main data sources can be established : OptiTrack and sonar.

4 Sonar and landing strategy

Now that the sun-position predicting model has been approved, the next part of the algorithm can be treated: the construction of an elevation map. To create this map, only 3D coordinates of the UAV are insufficient. In fact, the difference between the UAV z-coordinate and the sonar measurement is the height at a certain point. One of the limitations of sonar is usually noise, which is investigated before continuing with this approach. In Figure 5 the result of flying over a table is depicted.
There is definitely noise, but with a 2D discretized map these outliers could be canceled out. One should be careful in selecting a sonar sensor for this application though. With a damaged or malfunctioning sonar sensor extreme noise can be observed, certainly when operating close to operative engines with major vibrations.

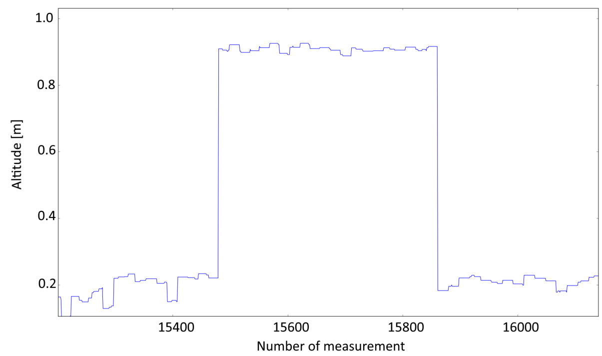


Figure 5: Sonar flight over table

Another aspect to consider is the fact that sonar gives point information. That is, it provides information about the environment at a certain point and not over a certain surface. This causes limitations which could be solved in the future with the use of computer vision.

4.1 UAV 3D coordinates

The other considerable input to compute an elevation map is the 3D coordinates of the UAV. This input is received from the OptiTrack system in TU Delft's Cyberzoo at the faculty of Aerospace Engineering. For outdoor operations, a GPS signal can be used.

4.2 2D discretized map

After sonar and position data has been retrieved, the continuous elevation map has to be discretized. A 2D grid with a known altitude for every cell is desired. In Figure 6 the left picture shows a discretized map and the right picture could be the original data.

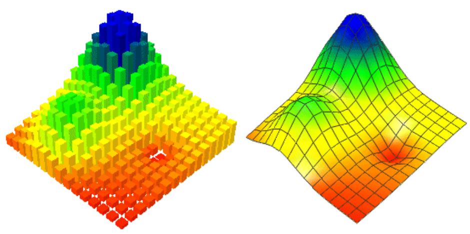


Figure 6: Elevation map discretization [11]

But why would destroying information be favourable? The most important reason is the point-information character of sonar. In reality, the UAV will not be able to cover every single centimeter with little coverage of the total area as a result. Sonar will actually create strings of information in the shape of the flight path, which is not the desired outcome. Instead, the measurements in each cell are averaged and this averaged value is used to apply the sun model. This process could be seen as converting strings of information to small surfaces of information.

4.3 Sun model application on environment

Once the elevation map has been discretized, the sun position predicting model can be applied. The most illuminated landing position over an entire charge period is to be found. For now charging time is assumed to be 24 hours, but this could easily be changed for different purposes.

4.4 Laser beam approach

The computation is relatively straightforward. The sun model outputs a finite amount of sun positions, for all these positions it is checked whether a virtual "laser beam" cuts through the environment before it leaves the grid. If it does not, for this sun position, this cell "sees" sun. This method is demonstrated in Figure 7.

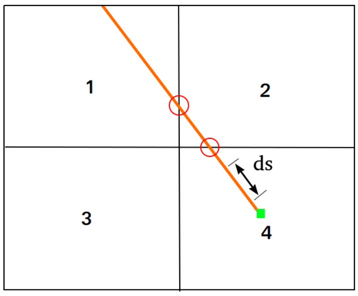


Figure 7: Laser beam approach

This is a test for cell four, where it starts in the middle of the cell at the green dot. The next step is walking along the "laser beam" over a distance ds per step. X, Y and Z are updated according to Equations 11, 12 and 13.

$$x_{i+1} = x_i + ds \cdot cos(\gamma_s) * cos(\alpha_s) \tag{11}$$

$$y_{i+1} = y_i + ds \cdot sin(\gamma_s) * cos(\alpha_s) \tag{12}$$

$$z_{i+1} = z_i + ds \cdot sin(\alpha_s) \tag{13}$$

Here i denotes the i-th timestep, α_s is solar elevation angle and γ_s is the solar azimuth angle.
Limitation of this model is that atmospheric conditions and solar cell orientation are not taken into account. Basically, the most illuminated landing position is calculated rather than the position with most solar energy.

4.5 UAV implementation

An efficient way to implement this algorithm into the Paparazzi autopilot software [12] has to be established. Two modules are added : one to collect data into arrays and one to analyze the data.

4.5.1 Data acquisition module

This module is fairly straightforward : it keeps loading sonar and position data into the desired arrays as long as the scanning pattern is not completed. These arrays are included in the header file of this module to later pass them to the analysis module.

4.5.2 Data analysis module

This module is somewhat more complex and consists of three parts, being : (i) Discretize obtained data, (ii) Compute finite amount of solar position and (iii) Apply model and choose best landing position.

The discretization is nothing more than taking the average of all measurements within one cell. After this has been done, the sun positions will be computed for a finite amount of points in time, depending on the computational power available.

Finally, the sun model needs to be applied. The general approach here is to leave the mesh relatively coarse at first to damp outliers as sonar has the tendency to show peaky outliers. Out of all cells, the mesh with highest solar intensity and with the highest altitude is chosen. Most important reason for this is that a high position is generally favourable in terms of sun, as there are less obstacles preventing sun from reaching that position. It could very well be that a large obstacle falls out of the scope of the scanning pattern and goes unnoticed. Therefore, it is useful to select the highest and stable landing position out of all positions with optimum illumination.
Then, after this cell has been chosen, it is decided on which place in that cell the UAV will land. This is done by looking at the original data and "walking" over this data in time. The landing position is chosen at the point in time where the area under the altitude graph between t-x and t+x is largest, that is, for Equation 14 :

$$P(t) = \int_{t-x}^{t+x} (z(t) - s(t))dt \tag{14}$$

where the highest value of P is obtained. Here x is in seconds and determines the time window considered, z(t) is the z GPS (OptiTrack) coordinate and s(t) is the sonar value. The advantage of this approach is that relatively easily the middle of a flat object can be found, as for example a table.
More in general, this two-step approach to first use a coarse mesh and than analyze data within that mesh makes for precise landing while still limiting noise. Only disadvantage is that a UAV must fly over the final landing position to finally land there, as no interpolation is done. Thisis a future step if sonar continues to be used, but computer vision is preferred.

As an example, imagine flying over a table. As shown in Figure 8 and 9, the middle of the table can be identified using Equation 14. The UAV will then land exactly there where it measured the maximum value of $P(t)$.

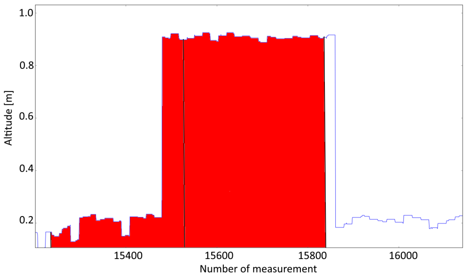


Figure 8: Table not centered

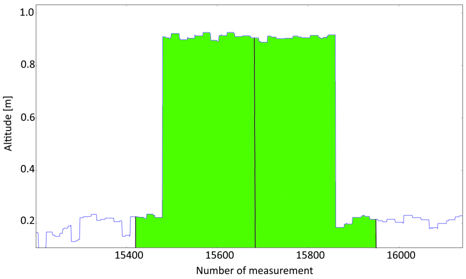


Figure 9: Table centered : max area

Using a denser mesh and analyzing multiple cells has also been considered. Anyhow, this approach did not damp out outliers enough. More importantly, scanning coverage was too limited for this approach. Only a limited amount of cells would be filled with data and the analysis becomes unreliable. Without interpolation, a cell without data would have an altitude of zero which would make the analysis a lot less reliable. This is why, at this stage, it is chosen to analyze a coarse mesh and in future work use computer vision with optic flow.

5 Flight experiments

Once an algorithm has been developed, it can be tested in flight experiments. One component of the algorithm is not yet constructed: the flight plan. The flight plan is crucial for a successful experiment and it is a continuous trade off: by scanning an environment in more detail more battery is sacrificed to recharge efficiently. On a huge flat surface this may be very inefficient, but in more populated areas it could be useful to take more time for scanning.
The general shape can be varied, but the two proposed shapes are zigzag and spirals. Zigzag means the UAV tries to scan a square by flying in U-shapes. That is, it crosses the square perpendicular, moves slightly to the side and crosses it again perpendicular. This is very similar to a creeping line search. Spirals are as the name implies: just spirals within the same square. These shapes could be used for environments where little is known and a more random flight plan is desired.
Both flight plans are programmed into Paparrazi open source autopilot [12].
After carefully constructing the algorithm in the framework of Paparazzi autopilot [12], a real test of this algorithm is to be performed. Main goal is to verify precision as well as plausibility of the outcome. Therefore, different test setups were constructed in TU Delft's Cyberzoo, as depicted in Figures 10 and 11.

Figure 10: Test setup 1 : small table on flat surface

Figure 11: Test setup 2 : random chairs added

The UAV should land on the table as it is the highest, flattest and most illuminated surface in the Cyberzoo. The small table demonstrates precision as it is a relatively small target. Besides, the chairs in Test setup 2 are added to test a more realistic environment.

5.1 Test Setup 1

After having performed the scanning pattern, all data is discretized on-board in a 10x10 mesh. The outcome of this discretization is visible in Figure 12 where the can be distinguished easily.
After the discretization, the local solar intensity was computed as visible in Figure 13. One might notice ”gaps” in solar intensity, which is caused by limited coverage by the

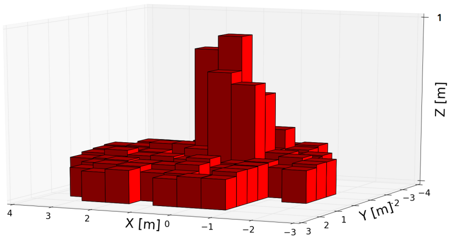


Figure 12: Test Setup 1 discretized

zigzag flightplan. Certain cells don't contain any data and are therefore assigned with a solar intensity score of zero (they are blue).

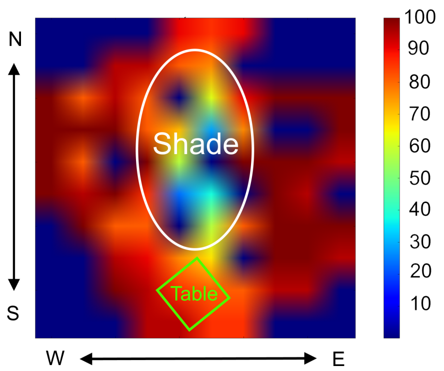


Figure 13: Solar intensity Test Setup 1 over a day, as % of the max

The final outcome in test setup 1 is that it landed on the table multiple times. However, using a well-suited flightplan was critical. It occasionally landed on the edge of the table, as that is where it flew straight over. Once it flew over the table straight, the UAV could land on the table.

5.2 *Test Setup 2*

Once setup 1 was relatively robust, a more complex environment was used for a test, depicted in Figures 14 and 15. Chairs were added randomly in different places.

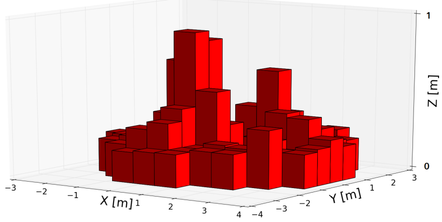


Figure 14: Test Setup 2 discretized

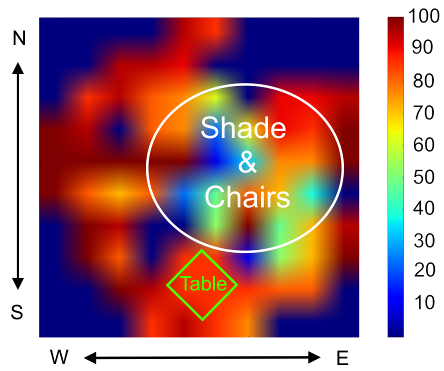


Figure 15: Solar intensity Test Setup 2 over a day, as % of the max

The UAV landed on the table multiple times again correctly but an appropriate flight plan remained important.

6 Feasibility

After the algorithm has been implemented in a Parrot Bebop 1 drone, it is useful to evaluate real world applications. With a preliminary estimation on charging capabilities of the DelftaCopter [13], more insight can be obtained on how feasible the idea is and what a possible recharge time would be. The DelftaCopter [13] is a delta-wing electric long range transitioning autonomous helicopter.

6.1 *Approach*

An accurate estimation of recharging performance is hard, as various factors play a major role. Among others, the landing area influences recharging performance by reflection and possible shades. For this feasibility study, it is estimated what order of magnitude the recharge time would be for a DelftaCopter with solar arrays on its wings. This is done by starting from the average received solar energy per day and with this compute the time required to recharge the DelftaCopter.

6.2 *Solar Energy*

For this, it is of great use to evaluate solar energy on a global level. Figure 16 is a map showing solar energy captured over the scope of a day (averaged) and a year respectively. It can be seen that Africa and Australia receive relatively high amounts of solar energy, while northern Europe is less suitable for the recharging application.

6.3 *Recharging the DelftaCopter*

Now that information on solar energy has been gathered, an estimated recharge time for the DelftaCopter can be computed. First of all, a solar cell is chosen: a Flexible Mono Solar Cell. Main advantages are its price, flexibility and efficiency. Its properties are visible in Table 1.

It has to be estimated now how many panels can be fitted on the DelftaCopter. Via straight forward geometric computations it can be computed that 7 of these cells can be

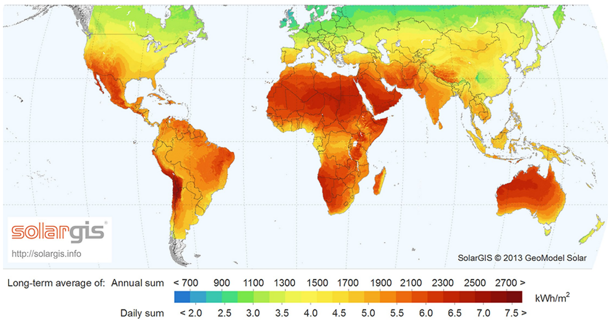


Figure 16: Global Solar Energy, SolarGIS ©2014 GeoModel Solar

Table 1: Flexible Mono Solar Cell 125x125 Monocrystalline, Amazon (25-7-2018)

Spec	Value
Efficiency	22.5 %
Size	125mm x 125mm
Power output	3.2-3.4 watts
Price for 10 pieces	50.5 USD

placed on one side of the wing and thus a stacking factor of 0.44 can be achieved.

It was stated before that this feasibility study will need assumptions and that the final computations will be a rough estimate. It is assumed now that only one side of the DelftaCopter is filled with the solar cells and that they take up 22.5 % of the solar energy they receive. This is similar to laying the drone on the ground horizontally, which would need hardware modifications in the current version of the DelftaCopter. Anyhow, another option would be to stack solar arrays along the other wing too to capture reflected sunlight. The exact layout is beyond the scope of this study and we will for now assume that the DelftaCopter is able to position the wing so that it lies flat on the ground.
To compute the recharge time, Equation 15 is used.

$$T_{charge} = \frac{E_{bat}}{n_{arrays}\, E_{sun}\, A_{array}\, \eta_{array}} \quad (15)$$

Here T_{charge} is the charging time in days, E_{bat} the energy available in the batteries in Wh, n_{arrays} the number of solar arrays used, E_{sun} the solar energy received over one day on 1 m^2 in Wh, A_{array} the surface area of 1 solar cell and finally η_{array} the efficiency of the solar cell.

Knowing that the approximate battery size of the DelftaCopter is 227 Wh and the solar cell properties from before, charging estimates for different positions around the globe can be performed. The results are shown in Table 2.

Again this is a very rough estimate, but it shows that recharging in 'a couple of days' is definitely feasible. Alternative configurations with folding mechanisms for solar arrays can be considered and make the solution more attractive with a lower recharge time.

Table 2: Recharging parameters around the globe

E_{sun} [Wh/m^2]	T_{charge} [days]	Avg. P_{cell} [W]	Region
7000	1.32	2.05	Australia, Africa
4500	2.05	1.32	USA, Asia
3000	3.07	0.88	Europe

7 Conclusion

Several studies have shown that solar arrays on the wing of an UAV during flight can be effective. Very little was found in literature on the question if using solar arrays to charge while standing on the ground could also be effective. The current study found that this can indeed be an effective way to substantially increase range for UAVs used for long-duration autonomous operations.
The contribution of this study has been to confirm that it is feasible to use this approach for UAVs as the DelftaCopter and make new types of missions possible without a return to home. Similarly, the algorithm developed in this study turned out to be successful and capable of finding the optimal landing position in several situations.
Interesting applications could be payload delivery or surveillance in vast remote areas. Mission duration might be extended, but cost and autonomous capabilities could be enlarged significantly.
UAVs will have to operate for an extended period of time in the future. This study is a step in the right direction on how to recharge effectively and enlarge mission capabilities to put UAVs to good use in a more effective and efficient way.

8 Future Work

The algorithm is limited to sonar and position data and needs more work to make the outcome more reliable and efficient. Vision based elevation mapping could be a major addition to the algorithm, as this allows for scanning of strokes instead of lines. Apart from that, a merge with other landing algorithms could be performed. Also, more work can be done on novel UAV configurations for this purpose. Folding solar arrays could be a solution. Finally, intelligent path planning for this application is another interesting topic. When and where to charge most effectively is an important topic, flying by night might be most efficient for example.

Acknowledgements

This study would not have been possible without the help of our colleagues in the MAVLab, who were extremely helpful. The facilities in the MAVLab, such as the Cyberzoo and the Bebop 1 used for the flight experiments have also been essential.

References

[1] Murat Bronz, Jean Marc Moschetta, Pascal Brisset, and Michel Gorraz. Towards a long endurance mav. *International Journal of Micro Air Vehicles*, 1(4):241–254, 2009.

[2] L. Wirth, P. Oettershagen, J. Ambhl, and R. Siegwart. Meteorological path planning using dynamic programming for a solar-powered uav. In *2015 IEEE Aerospace Conference*, pages 1–11, March 2015.

[3] H. Bahrami Torabi, M. Sadi, and A. Yazdian Varjani. Solar power system for experimental unmanned aerial vehicle (uav); design and fabrication. In *2011 2nd Power Electronics, Drive Systems and Technologies Conference*, pages 129–134, Feb 2011.

[4] Steven Anton. Energy harvesting for unmanned aerial vehicles. 01 2008.

[5] Guoqing Zhou and Deyan Zang. Civil uav system for earth observation. In *2007 IEEE International Geoscience and Remote Sensing Symposium*, pages 5319–5322, July 2007.

[6] Nicola Casagli, William Frodella, Stefano Morelli, Veronica Tofani, Andrea Ciampalini, Emanuele Intrieri, Federico Raspini, Guglielmo Rossi, Luca Tanteri, and Ping Lu. Spaceborne, uav and ground-based remote sensing techniques for landslide mapping, monitoring and early warning. *Geoenvironmental Disasters*, 4(1):9, Mar 2017.

[7] Mingxing Gao, Xiwei Xu, Yann Klinger, Jerome van der Woerd, and P Tapponnier. High-resolution mapping based on an unmanned aerial vehicle (uav) to capture paleoseismic offsets along the altyn-tagh fault, china. 7, 12 2017.

[8] G.C.H.E. de Croon, H.W. Ho, C. De Wagter, E. van Kampen, B. Remes, and Q.P. Chu. Optic-flow based slope estimation for autonomous landing. *International Journal of Micro Air Vehicles*, 5(4):287–297, 2013.

[9] Omid Shakernia, Yi Ma, T. John Koo, and Shankar Sastry. Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control. *Asian Journal of Control*, 1(3):128–145.

[10] A. A. Rizvi, K. Addoweesh, A. El-Leathy, and H. Al-Ansary. Sun position algorithm for sun tracking applications. In *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, pages 5595–5598, Oct 2014.

[11] Mark Dorey Derek Smith David Kidner. What's the point? interpolation and extrapolation with a regular grid dem. *GeoComputation 99*, 1999.

[12] Pascal Brisset, Antoine Drouin, Michel Gorraz, Pierre-Selim Huard, and Jeremy Tyler. The Paparazzi Solution. In *MAV 2006, 2nd US-European Competition and Workshop on Micro Air Vehicles*, page pp xxxx, Sandestin, United States, October 2006.

[13] Christophe De Wagter, Rick Ruijsink, Ewoud Smeur, Kevin van Hecke, Freek van Tienen, Erik van der Horst, and Bart Remes. Design, control and visual navigation of the delftacopter. *CoRR*, abs/1701.00860, 2017.

Enabling Intelligent and Autonomous Drones Using Embedded Linux

K. Amancio*, G. Pinheiro, L. Teixeira and C. Duarte
Federal University of Itajubá, Av. B P S 1303, Itajubá

ABSTRACT

In the current scenario, drones became more than simple toys for hobbyists and started to be tools with many applications. Among those, there are natural disaster prevention and combat, surveillance and monitoring, environmental and civil inspection, public security, rescue and salvage, among others. In many of those applications it is desirable that the aircraft operates autonomously, aiming at greater efficiency in the execution of those tasks. In this article, it will be shown how the technology of embedded computers running a Linux distribution as operating system can make possible to achieve a level of autonomy and intelligence within these aircrafts.

1 INTRODUCTION

This article aim to demonstrate how it is possible to achieve intelligent and autonomous drones using the technology of embedded computers running a Linux distribution as operating system and communicating with a flight controller board on a quadcopter aircraft.

2 ESTABLISHING COMMUNICATION BETWEEN FLIGHT CONTROLLER AND COMPANION COMPUTER

For the purpose of having intelligent and autonomous vehicles for various tasks, first is necessary to integrate a so called companion computer on the drone. These devices often are single board computers, which are credit card size boards with some computational resources like a microprocessor running an Operating System, memory, graphic unit and input/output (I/O).

In this article, we adopted the DragonBoard™ 410c as the companion computer, due to his features as listed below.

- Quad-core ARM® Cortex® A53 at up to 1.2 GHz per core with both 32-bit and 64-bit support
- 1GB LPDDR3 533MHz / 8GB eMMC
- Qualcomm Adreno 306 GPU with support for advanced APIs,
- 1080p@30fps HD video playback and capture
- Wi-Fi 802.11 b/g/n 2.4GHz, integrated digital core
- Bluetooth 4.1, integrated digital core
- On-board Wi-Fi, BT and GPS antenna
- Two USB 2.0 and a micro SD card slot
- One 40-pin low speed expansion connector: UART, SPI, I2S, I2C x2, GPIO x12 and DC power.

On the vehicle, we have a PixHawk Flight Controller Board running a Flight Stack, software and firmware responsible for the stability of the aircraft.

2.1 Physical Connection

To make PixHawk and Dragonboard communicate, we use the UART protocol, available through the

*contact email: kelvin013@gmail.com

TELEM1 port on PixHawk and through UART0 pins on Dragonboard.

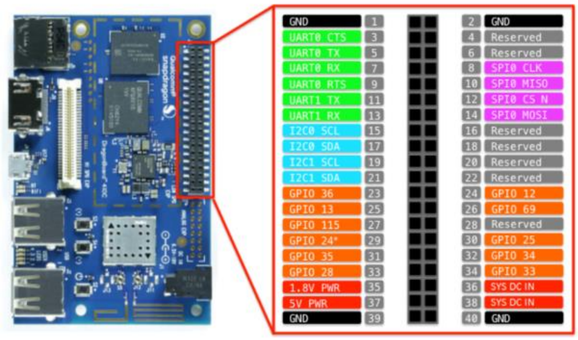


Figure 1 – Dragonboard 410c Pinout

TELEM1, TELEM2 ports

Pin	Signal	Volt
1 (red)	VCC	+5V
2 (blk)	TX (OUT)	+3.3V
3 (blk)	RX (IN)	+3.3V
4 (blk)	CTS	+3.3V
5 (blk)	RTS	+3.3V
6 (blk)	GND	GND

Figure 2 – Telemetry Ports on PixHawk

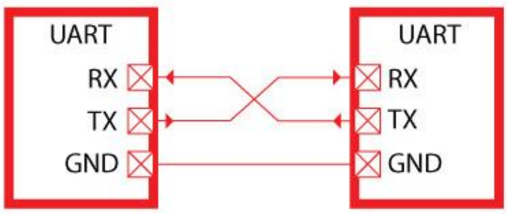


Figure 3 – Connection between Devices Using UART Ports

2.1 Logical Connection

Furthermore, these two devices also need one protocol to establish a connection. This is achieved with MAVLink (Micro Air Vehicle Communication Protocol) that runs on both systems.

As soon as the two devices are properly configured with the correct requirements and libraries and the two boards are physicaly connected, the connection can then be established.

```
SERIAL2_PROTOCOL = 1

SERIAL2_BAUD = 921

LOG_BACKEND_TYPE
```

Figure 4 – Parameter that Need to be Modified in the PixHawk Settings

```
$ sudo apt update

$ sudo apt install screen python-wxgtk3.0
python-matplotlib python-opencv
python-pip python-numpy
python-dev libxml2-dev libxslt-dev
python-lxml

$ sudo pip install future

$ sudo pip install pymavlink

$ sudo pip install mavproxy
```

Figure 5 – Linux Commands to Install Necessary Libraries and Dependencies on Dragonboard

```
$ mavproxy.py --master=/dev/ttyMSM1
--baudrate=57600
```

Figure 6 – Linux Commands to Test the Connection between Dragonboard and PixHawk

If the connection is successfully established, then it is now possible to run algorithms developed in a programming language such as python to command the aircraft to perform a diversity of tasks, such as avoid collision, detect a target, identify and follow an object, among many others.

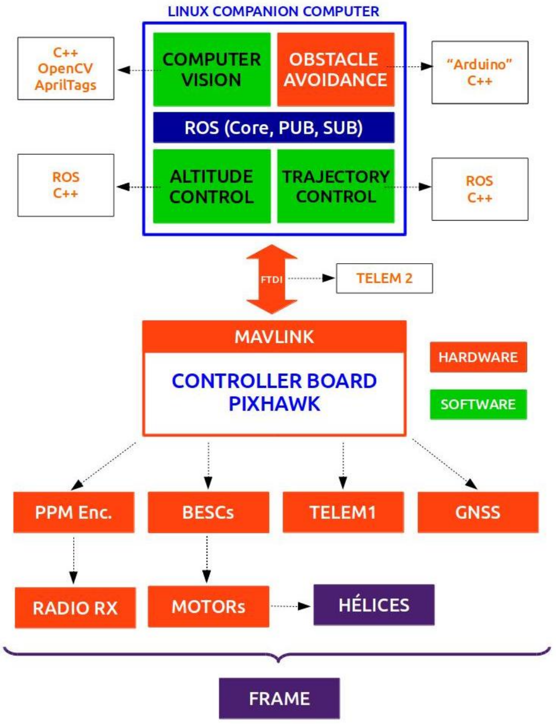


Figure 7 – Drone Schematic Representing the Devices on the Aircraft and Applications that can be Executed on the Companion Computer

3 ROS AND COMPUTER VISION

On the Black Bee Drones team, we have a division called Software, in that division, we develop the artificial intelligence that is applied to our drones. We do that using ROS (Robot Operating System) and Computer Vision Algorithms to recognize the environment, understand the data obtained and make decisions based on that.

Our primary tool that makes all of this autonomy possible is ROS (Robotic Operating System). ROS is a meta-operating system that allows us to work with a heterogeneous group of computers to control our drone from an onboard computer or a ground station where we apply our computer vision algorithms so that the environment around the drone is understood and a decision can be made.

The decision-making process is divided into two steps, first, all the data from the drone's camera is read, this data is processed by the algorithms developed by our team with a strong focus on specific tasks, like finding an object or pattern.

The second step is when those algorithms are used to recognize answer if something where found or not. The control part of the decision-making is then activated; this is when the drone executes an action based on the results obtained from the data processing. For example, an algorithm designed to find and follow an object; the drone would get the coordinates of the object on relation to itself through the camera and follow the movement based on the variation of where the object is on the image obtained.

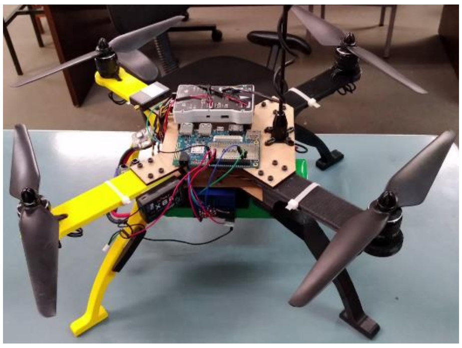


Figure 8 – Drone with PixHawk and Dragonboard Communicating Through UART Connection

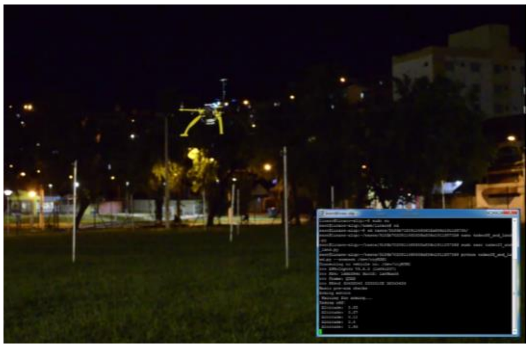


Figure 9 – Take Off, Rover and Land Test with a Python Algorithm Running on Dragonboard

4 CONCLUSION

After all the procedure described in this article, it is demonstrated that is possible to obtain aircraft with the capability of execute autonomous tasks, such as avoid collision, target detection, identification and following an object. This improves the efficiency, flight autonomy and security in comparison with a manual controlled aircraft and the development of this technology is the goal of Black Bee Drones team.

ACKNOWLEDGEMENTS

We would like to express our deep gratitude to Professor Dr. Alexandre Carlos Brandão Ramos, our project supervisor, and all of ours Black Bee Drones colleagues.

REFERENCES

FLOREANO, D.; WOOD, R. J. Science, technology and the future of small autonomous drones. Nature, Nature Research, v. 521, n. 7553, p. 460–466, 2015.

MEIER, L. et al. Pixhawk: A system for autonomous flight using onboard computer vision. p. 2992–2997, 2011.

Saif Aldeen Saad Obayes Al-Kadhim and Imam Alkadhim Collage. Communicating with raspberry pi via mavlink. 2016.

Towards Intelligent Aircraft Through Deep Reinforcement Learning

S. Morrison*, A. Fisher, and F. Zambetta
RMIT University, 414-418 Swanston St, Melbourne, VIC, 3000

ABSTRACT

Deep reinforcement learning has achieved recent successes in solving games and learning robotics tasks from scratch, and has shown early promise for the guidance, navigation, and control of MAVs. Though MAV control is well-established, many complex tasks still require human oversight, and techniques for reducing the level of human involvement are still nascent. In this paper, we present ongoing work in applying continuous-action deep reinforcement learning to autonomous aircraft in simulation, in order to learn such complex tasks autonomously. We provide a brief overview of our simulation environment and tasks of interest, and present preliminary results using model-free methods to learn simple flight tasks. We conclude with remarks on potential directions of research that we believe will have an impact on the future of unmanned systems.

1 INTRODUCTION

Deep reinforcement learning is a framework for training controllers in a manner that mimics learning in biological organisms [1,2]. It has recently achieved critical successes in playing computer Go [3,4], computer games [5,6], and learning nonlinear controllers for tasks such as robot locomotion and grasping [7,8]. Preliminary results have also shown that reinforcement learning can be used to train MAV flight controllers capable of complex flight tasks such as aircraft recovery [9] and collision avoidance [10]. Given RL's ability to learn complex we might hope to use it to learn robust flight controllers from scratch that enable greater levels of autonomy.

One of the primary impediments to this goal is the lack of a framework for training aircraft controllers. Deep reinforcement learning relies on the existence of simulated environments to train policies; applications of RL to the MAV domain in the literature are split on the use of the Robot Operating System (ROS) with Gazebo (typically RotorS [11]), and custom flight simulations. This is at odds with much of RL research, for which the primary benchmark is the OpenAI Gym framework. For our research, we are interested in learning individual flight tasks, and utilizing existing RL implementations. Whilst frameworks like RotorS are excellent for standard robotics research, they are unsuitable for RL, for which we might want to trial many thousands of episodes across a range of environments, in order to learn a policy.

In this paper, we present on-going work in building such a framework for learning nonlinear quadrotor flight controllers. Our framework extends OpenAI Gym, and includes flight control tasks such hovering, random waypoint navigation, landing, and target following. We hope that by sharing it, we can spur further innovation and development of intelligent flight controllers for small unmanned systems.

2 BACKGROUND

2.1 Deep Learning

Deep learning is a function approximation method loosely based on neural networks in the brains of biological organisms [1,2]. An artificial neuron takes input from surrounding neurons, performs a nonlinear transformation (known as an activation function), and then passes this output as input to a group of connected neurons [2]. The typical structure of a neural net is given in Figure 1, and

*contact email: address@domain.com

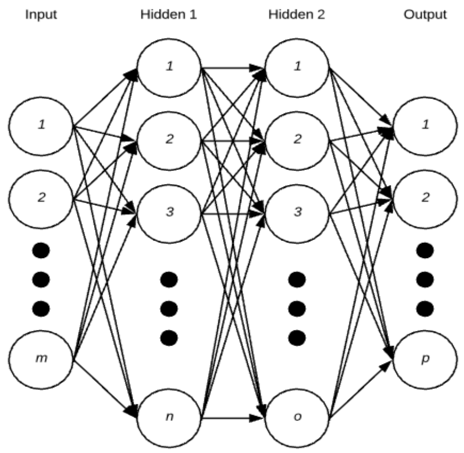


Figure 1 – Typical Multi-layer Perceptron

consists of a series of sequentially connected layers known as a Multilayer Perceptron (MLP). Neurons are connected to one another via a group of weights $\theta \in \Theta$, with the process of learning strengthening important weight connections and weakening detrimental ones. At a simple level, a neural network can approximate basic functions such as polynomials, but their real strength lies in being able to approximate functions that cannot be represented in any other way – for example, a function that detects cats in images, or estimates the value of an action – by learning them. The network is trained by minimizing a cost function or maximizing a score function. Gradients for each weight are pushed backwards through the network using the backpropagation algorithm – a variant of the chain rule from differential calculus – and an optimizer is used to step the weights in the given direction.

Specialist architectures such as Convolutional Neural Networks (CNNs) can learn filters that aid in the detection and classification of objects, and Recurrent Neural Networks (RNNs) are able to model dependencies through time [2]. CNNs see use in image classification, and have been applied to reinforcement learning for games. RNNs are used in cases where information is relevant over long time scales, and maintain a short term memory that aids in learning temporal dependencies [2]. Recent work has seen the development of generative models that learn to synthesize new examples of their training data [12,13], and memory networks that extend the functionality of RNNs through the use of an external long-term memory module [14,15]. These newer techniques are now finding their way into natural language processing and deep reinforcement learning, where they are learned to 'imagine' new scenarios and remember information over long very time scales [16,17].

2.2 Reinforcement Learning

Reinforcement learning (RL) deals with the problem of training an agent to act intelligently in an environment using an external reward signal [18]. The agent begins in an initial state s_0, and takes an action a_o according to a policy π that determines how actions should be selected. The agent receives a reward r_0 from the environment, and arrives in a new state s_1 that is governed by a transition probability $P(s_{t+1}|s_t, a_t)$. We refer to a sequence of such events as a trajectory (denoted τ) where $\tau = \{s_0, a_0, r_0, s_1, \dots, s_{t-1}, a_{t-1}, r_{t-1}, s_T\}$. The goal of an agent acting in an environment is take a trajectory that maximizes the total expected reward [18]:

$$R_t = \mathbb{E}_\pi \left[\sum_{k=0}^{T} \gamma^k r_{t+k+1} \right] \tag{1}$$

Where $\gamma \in [0,1]$ is a discount factor that more heavily weights immediate rewards, and ensures that the sum converges for the horizon $T = \infty$.

RL assumes a Markov Decision Process (MDP), meaning that the past and present are conditionally independent of one another given the present. More concretely, all information necessary for taking the best action is assumed to be included in the present state of the agent. As an example, a robot with a battery can take two trajectories, and end up in the same state s_t at the same time t, with different levels of remaining charge. If the battery's charge is not included in the robot's state, the Markov property is violated, and learning becomes more difficult (for example, the robot may not have enough remaining charge to reach the goal). In this case, the Markov property can be preserved by ensuring the robot can "see" its current level of charge.

RL problems are typically solved by learning a value function $V: S \rightarrow \mathbb{R}$, that maps the relative value of each state (known as the state value function for π, $V^{\pi}(s_t)$), or each state-action pair $Q: S \times A \rightarrow \mathbb{R}$ (known as the state-action value function for π, $Q^{\pi}(s_t, a_t)$). Actions are selected according to the policy π; for example, our policy might be to take the highest value action with probability P, and choose a random action with probability $1 - P$. Similarly, if we know the true value of every state, we can always choose the highest value actions (known as the greedy action). Our policy can also be a function $\pi: S \rightarrow A$ that maps states directly to actions. This function is often a probability distribution over actions $\pi_{\theta}(a_t|s_t)$ with arbitrary parameters θ, with learning being the task of optimizing θ. This can be done by sampling actions from $\pi_{\theta}(a_t|s_t)$, evaluating their quality, and then adjusting θ accordingly.

Since – in continuous state-and-action spaces – there are infinite possible values, modern applications use function approximation to learn $V^{\pi}(s_t)$ and $\pi_{\theta}(a_t|s_t)$. For $V^{\pi}(s_t)$, this is done using supervised learning, by first rolling out an episode, and then calculating the return at each state using Equation 1. A cost function is minimised using these values as a target. The policy can be trained in multiple ways, but one of the more common techniques – known as Monte-Carlo Policy Gradient – uses a score function estimator of the form:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T-1} \nabla_{\theta} \log \pi_{\theta}(a_t|s_t)\, \Phi_t\right] \qquad (2)$$

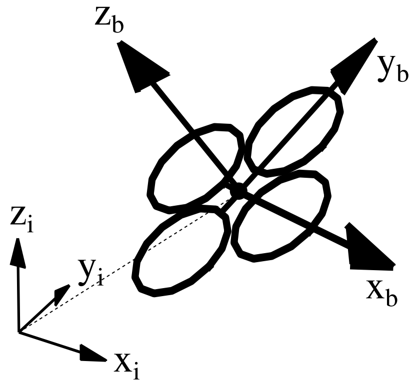


Figure 2 – Aircraft axis system

Where π is parameterized by weights θ, and Φ_t can take multiple forms, but is typically some variant of Equation 1 (see [18,19,20] for derivation and exposition of the Policy Gradient Theorem). By stepping the policy weights in the direction that maximizes the value function, it can progressively be trained to take better actions. The expressive power of neural networks makes them a popular choice for representing both the value function and the policy; when deep learning is combined with reinforcement learning, the combination is known as deep reinforcement learning.

3 SIMULATION AND ENVIRONMENTS

We model a plus-configuration aircraft in an East-North-Up axis system, and assume a flat Earth with constant air density. A diagram of our aircraft's axis system is given in Figure 2. Our equations of motion are:

$$\begin{bmatrix} 0 \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} 0 \\ \dfrac{\boldsymbol{F}_T + \boldsymbol{F}_A}{m} \end{bmatrix} + \boldsymbol{q} \begin{bmatrix} 0 \\ \boldsymbol{G}_i \end{bmatrix} \boldsymbol{q}^{-1} - \begin{bmatrix} 0 \\ \boldsymbol{\omega} \times \boldsymbol{v} \end{bmatrix} \qquad (3)$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{J}^{-1}(\boldsymbol{M}_T + \boldsymbol{M}_A - \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega}) \qquad (4)$$

$$\dot{\boldsymbol{x}} = \boldsymbol{q}^{-1} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{v} \end{bmatrix} \boldsymbol{q} \qquad (5)$$

$$\dot{\boldsymbol{q}} = -\frac{1}{2} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\omega} \end{bmatrix} \boldsymbol{q} \qquad (6)$$

Where $\boldsymbol{x}$ is the position vector in the inertial frame, $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are the linear and angular velocity vectors in the body frame, $\boldsymbol{F_T}$ and $\boldsymbol{F_A}$ denote the thrust and aerodynamic forces in the body frame, $\boldsymbol{M_T}$ and $\boldsymbol{M_A}$ denote the thrust and aerodynamic moments in the body frame, $\boldsymbol{G_i}$ is the gravity vector in the inertial frame, m is the mass of the aircraft, $\boldsymbol{J}$ is the inertia tensor, and $\boldsymbol{q}$ is a quaternion encoding the attitude of the aircraft. $\boldsymbol{q}$ is converted to Euler angles $\boldsymbol{\zeta}$, which are used by the RL environments. We use dot notation to denote the time derivative.

Thrust and torques are modelled as:

$$\begin{bmatrix} F_T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} k_T & k_T & k_T & k_T \\ 0 & lk_T & 0 & -lk_T \\ -lk_T & 0 & lk_T & 0 \\ -k_Q & k_Q & -k_Q & k_Q \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{7}$$

For scalar thrust and moment coefficients k_T and k_Q, and arm length l. Aerodynamic forces and moments are:

$$\boldsymbol{F}_A = -k_D \boldsymbol{v}^{\boldsymbol{T}} \boldsymbol{v} \widehat{\boldsymbol{v}} \tag{8}$$

$$\boldsymbol{M}_A = -k_M \boldsymbol{\omega}^{\boldsymbol{T}} \boldsymbol{\omega} \widehat{\boldsymbol{\omega}} \tag{9}$$

For scalar drag and aerodynamic moment coefficients k_D and k_M. Rotor speeds are modelled as a first order linear differential equation:

$$\dot{\boldsymbol{\Omega}} = -k_c(\boldsymbol{\Omega} - \boldsymbol{\Omega_c}) \tag{10}$$

that ensures the RPM adjusts smoothly over time with policy commands. We step the simulation forward using a standard RK4 integrator. We do not currently include more advanced effects such as ground effect, blade flapping, vortex ring state, or wind, though some of these are planned for future work.

Current environments include:

1. Hover, for which the aim is to hover on a static waypoint for the duration of an episode;
2. Static waypoint navigation, for which the goal is to sequentially navigate to a waypoint and then hover there for the remainder of the episode;
3. Random waypoint navigation, for which the goal is to navigate to a randomly generated waypoint within a given distance to the aircraft;
4. Flying straight and level, for which the aircraft must fly in a constant direction for as long as possible, whilst maintaining constant altitude;
5. Landing, for which the aircraft must smoothly land without crashing, and without descending through its own rotor wash; and.
6. Target following, for which the aircraft must keep to a constant distance from a target as it moves through the environment.

Our reward function provides the aircraft with a positive reward for getting closer to a given goal, and a negative reward for moving away from it.

Our environments are episodic and terminate when the time limit has been reached or a termination condition has been met. These typically include flying beyond a given distance from the goal, though in the case of landing we also check for crashes. We ensure that the aircraft doesn't descend through its own rotor wash by checking the velocity in the aircraft's z-axis, and terminating the episode if it goes below -2m/s.

The observation space that is provided as input to the policy is often task dependent, but at a minimum it includes the position vector $\boldsymbol{x}$, $\sin(\boldsymbol{\zeta})$ and $\cos(\boldsymbol{\zeta})$, linear and angular velocities in the body frame $\boldsymbol{v}$ and $\boldsymbol{\omega}$, and – in goal-conditioned cases – the vector pointing towards the goal, $\boldsymbol{g}$. The target following case also includes the

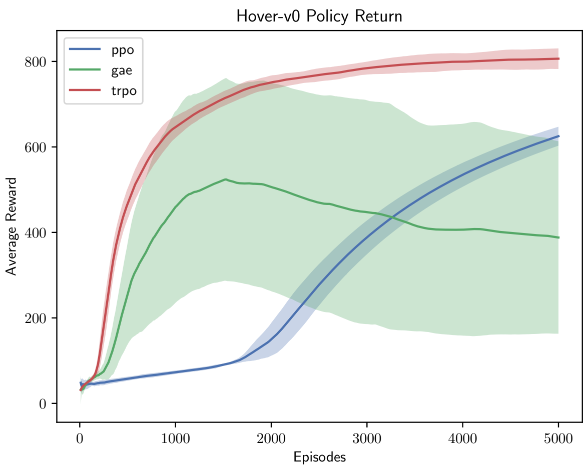


Figure 3 – Learning curves for the Hover task

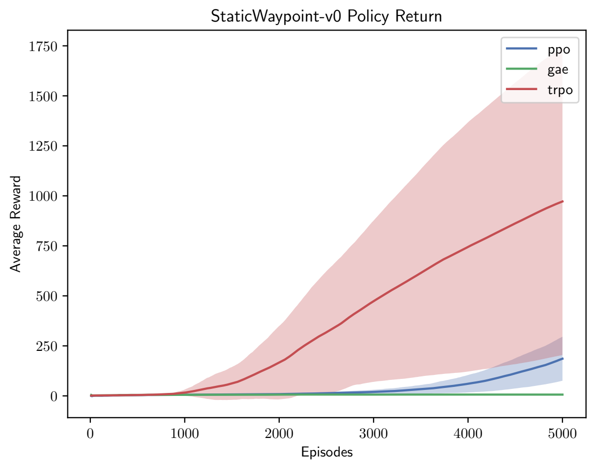


Figure 4 – Learning curves for the Static Waypoint task

velocity vector of the target. Per the MDP assumption, this information needs to be included in the observation space, and is realistically attainable for an aircraft. As our simulation does not currently include wind, we don't differentiate between the ground speed and airspeed of the aircraft. This is expected to change in future updates.

4 EXPERIMENTS

We applied three state-of-the-art Monte-Carlo policy gradient algorithms to learning hover, static waypoint navigation, and landing tasks. The algorithms we use are Generalized Advantage Estimation (GAE), Proximal Policy Optimization (PPO) and Trust Region Policy Optimization (TRPO), respectively. For further insight into these methods, we refer the reader to [21,22,23].

Our policies are standard MLPs with 128 neurons in a single hidden layer, and learn a delta from the hover RPM. The networks output the mean and log-variance of the action, and we use this to construct and sample from a normal distribution over actions.

5 RESULTS

We show learning curves in Figures 3, 4 and 5, and trajectory plots are given in Figures 6 and 7.

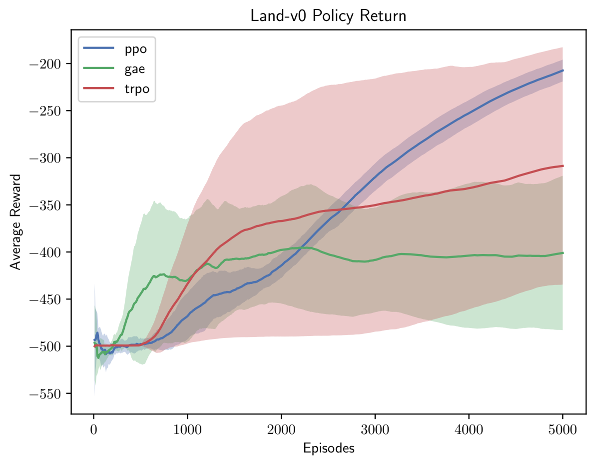


Figure 5 – Learning curves for the Landing task

We found that PPO and TRPO produced the most consistent overall performance, and provide video footage of our controllers at [address withheld]. In general, stochastic policies don't produce analogous performance to a modern controller, since they sample from a distribution. A stochastic policy will always produce a slight wobble due to its probabilistic nature. An advantage of this is that such policies should be robust to additional disturbances such as wind, and can produce smoother behaviour by taking the mean action, rather than sampling from the distribution.

In practice, we found that we were able to learn policies capable of achieving most goals. We

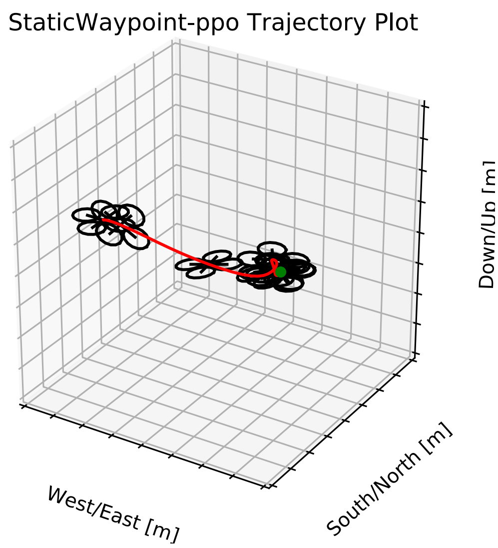


Figure 6 – Static Waypoint trajectory plot

found PPO to be the most consistent algorithm, though TRPO learned to flight behaviour that was visually smoother and more controlled. This is likely due to the difference in optimization methods – TRPO is a second order algorithm, whereas PPO is a first order method that approximates TRPO. PPO uses an optimizer known as ADAM that has been shown to find sharper local minima than other methods. We believe this is a potential underlying cause for the difference in flight behaviour.

Whilst GAE was able to learn most tasks, we found its performance to be higher variance than the other algorithms, and prone to collapse during training. This indicates that GAE might be less suitable for MAV flight control applications.

6 FUTURE WORK

Future work will involve expanding on the current environments to include navigation through unknown domains, perching, and the inclusion of stochastic wind. We aim to standardize the observation space across tasks so that we can use our current framework for multi-task learning (that is, a single controller that is capable of performing all tasks given some task parameter).

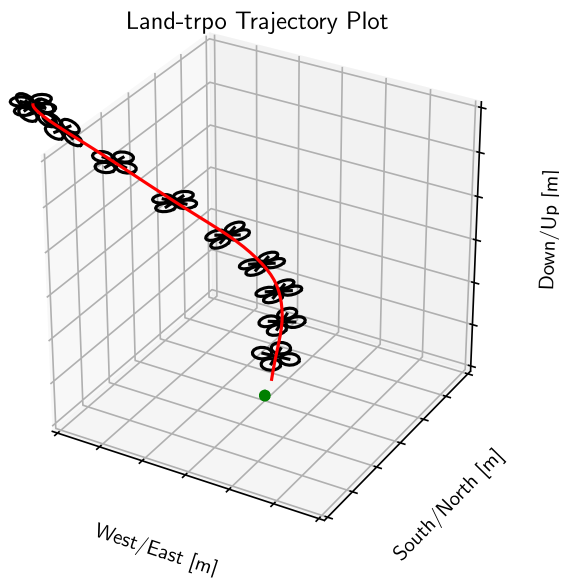


Figure 7 – Landing trajectory plot

Our goal is to fly a learned policy on an aircraft sometime towards the end of 2018.

6 CONCLUSION

The future of autonomous systems likely involves creating context-aware AI that is able to operate intelligently across a broad range of situations and tasks. The tools to build such systems might include recurrent neural networks that are able to make use of short term memory, memory-augmented networks that are able to write-to and read-from an external memory module, variational autoencoders that automatically learn a compressed representation of their input, and generative adversarial networks that can "imagine" new scenarios. Though such techniques are still difficult to train, research is ongoing and much preliminary work has already been done.

Training such agents is contingent on the existence of adequate simulation environments. We have presented ongoing work on such a framework that allows us to train up simple flight control policies, and we hope that our efforts may spur further effort in developing more intelligent flight controllers for MAVs.

ACKNOWLEDGEMENTS

The research was undertaken by the RMIT Unmanned Aircraft Systems Research Team, within the Sir Lawrence Wackett Aerospace Research Centre at RMIT University.

REFERENCES

[1] Russell, S.J. and Norvig, P., 2016. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited.

[2] Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y., 2016. *Deep learning* (Vol. 1). Cambridge: MIT press.

[3] Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. and Dieleman, S., 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, *529*(7587), p.484.

[4] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A. and Chen, Y., 2017. Mastering the game of Go without human knowledge. *Nature*, *550*(7676), p.354.

[5] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M., 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[6] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G. and Petersen, S., 2015. Human-level control through deep reinforcement learning. *Nature*, *518*(7540), p.529.

[7] Heess, N., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, A., Riedmiller, M. and Silver, D., 2017. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*.

[8] Arulkumaran, K., Deisenroth, M.P., Brundage, M. and Bharath, A.A., 2017. A brief survey of deep reinforcement learning. arXiv preprint arXiv:1708.05866.

[9] Hwangbo, J., Sa, I., Siegwart, R. and Hutter, M., 2017. Control of a quadrotor with reinforcement learning. IEEE Robotics and Automation Letters, 2(4), pp.2096-2103.

[10] Andersson, O., Wzorek, M. and Doherty, P., 2017, February. Deep Learning Quadcopter Control via Risk-Aware Active Learning. In *AAAI* (pp. 3812-3818).

[11] Furrer, F., Burri, M., Achtelik, M. and Siegwart, R., 2016. Rotors—A modular gazebo mav simulator framework. In *Robot Operating System (ROS)* (pp. 595-625). Springer, Cham.

[12] Kingma, D.P. and Welling, M., 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[13] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[14] Graves, A., Wayne, G. and Danihelka, I., 2014. Neural turing machines. arXiv preprint arXiv:1410.5401.

[15] Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S.G., Grefenstette, E., Ramalho, T., Agapiou, J. and Badia, A.P., 2016. Hybrid computing using a neural network with dynamic external memory. Nature, 538(7626), p.471.

[16] Zhang, J., Tai, L., Boedecker, J., Burgard, W. and Liu, M., 2017. Neural SLAM. *arXiv preprint arXiv:1706.09520*.

[17] Ha, D. and Schmidhuber, J., 2018. World Models. *arXiv preprint arXiv:1803.10122*.

[18] Sutton, R.S. and Barto, A.G., 1998. *Reinforcement learning: An introduction* (Vol. 1, No. 1). Cambridge: MIT press.

[19] Peters, J. and Schaal, S., 2006, October. Policy gradient methods for robotics. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* (pp. 2219-2225). IEEE.

[20] Peters, J. and Schaal, S., 2008. Reinforcement learning of motor skills with policy gradients. *Neural networks*, *21*(4), pp.682-697.

[21] Schulman, J., Moritz, P., Levine, S., Jordan, M. and Abbeel, P., 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

[22] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O., 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[23] Schulman, J., Levine, S., Abbeel, P., Jordan, M. and Moritz, P., 2015, June. Trust region policy optimization. In *International Conference on Machine Learning* (pp. 1889-1897).

Simulation and Flight Experiment of a Quadrotor Using Disturbance Observer Based Control

Hoijo Jeaong*[1], Seunghwan Jo†[1], Seungkeun Kim‡[1], Jinyoung Suk§[1], and Yung-Gyo Lee¶[2]

[1]Department of Aerospace Engineering, Chungnam National University, 99 Daehak-ro, Yuseong-gu, Daejeon 305-764, Korea
[2]Korea Aerospace Research Institute

ABSTRACT

This study constructs the realistic aerodynamic and wind database for a quadrotor and applies disturbance observer based control (DOBC). The wind-tunnel experiment measures six degree-of-freedom forces and moments in various combinations of wind velocity and wind directions -90°∼ 90°angle of attack and -45°∼ 45°sideslip. Computational fluid dynamics (CFD) at the urban field is applied to simulate a complex airfield environment in severe weather conditions. The simulator integrates a multidimensional lookup table to simulate different environments that include the location of a quadrotor, time, and the condition of wind. Then, the disturbance observer based controller is designed with the simplified longitudinal and lateral moment dynamics of the quadrotor to compensate the nominal controller based on conventional PID control. This study compares the performance of the disturbance observer based control with that of the PID controller through simulation and flight experiment.

1 INTRODUCTION

Quadrotor systems are susceptible to disturbance due to their trade-offs in mechanical simplicity compared to efficient aerodynamic design as well as the complexity of rotor systems. In particular, wind disturbance is an important issue for quadrotor safety and mission viability in the performance of complex urban terrain and bad weather condition. In this paper, a quadrotor aerodynamic database is constructed and a control strategy for omnidirectional disturbance is established by designing a disturbance observer. In a previous study on quadrotor modeling and simulation, quadrotor modeling was performed by calculating the thrust coefficient through motor specification and thrust measurements experiment [1, 2, 3]. However, these studies has limitations, it does not reflect the dynamic characteristics of the whole region of the quadrotor and does not reflect the model characteristics for controlling the disturbance. Therefore, for the precise modeling of the quadrotor in the simulation environment, this study constructes a quadrotor aerodynamic database based on the wind tunnel test. The wind tunnel experiment using DJI MATRICE 100 was conducted by the Korea Aerospace Research Institute (KARI). The aerodynamic test was performed under varying wind speeds and measured with three components of forces and moments and rpm. For more accurate modeling, we used the DJI MATRICE 100 CAD modeling data and motion capture cameras to measure the mass moment of inertia uses. In this paper, numerical simulation consists of multidimensional lookup tables for aerodynamics data provided by the KARI. The datadase consists of interpolated for a wide range of aerodynamic data. In this study, to compensate the limit of the PID controller, the DOBC is designed along with the PID controller on the pitch and roll axis respectively. Disturbance observer is a technique of observing or estimating model uncertainties and their effects on the system. The disturbance observer based control allows estimation by filtering the data of uncertainty such as the external load of the motor, friction force, and wind gust to compensate for the disturbance. In this study, we analyzed the performance of DOBC using numerical simulation. In addition, performance analysis using a blower fan was applied to the quadrotor by applying the DOBC. Finally, the performance of the disturbance observer was verified by calculating the RMSE (Root Mean Square Error) using the flight data acquired through the waypoint flight test.

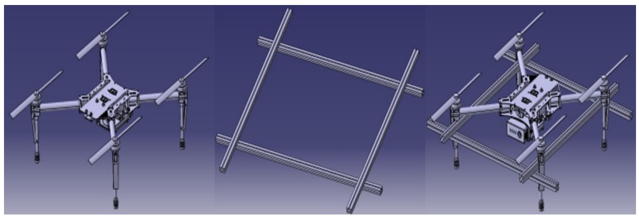


Figure 1: CAD modeling of DJI MATRICE 100 and apparatus.

*Email address(es): jeaonghoijo@cnu.ac.kr
†Email address(es): snipertype@cnu.ac.kr
‡Email address(es): skim78@cnu.ac.kr (corresponding author)
§Email address(es): jsuk@cnu.ac.kr
¶Email address(es): lyg@kari.re.kr

Table 1: DJI MATRICE 100 Specification.

Structure	
Diagonal wheelbase	650 mm
Weight	2355 g
Max takeoff weight	3600 g
Performance	
Max pitch angle	35 degree
Max speed of ascent	5 m/s
Max speed of descent	4 m/s
Max wind resistance	10 m/s
Max speed	17 m/s (no Wind)
Hovering time	22 min

Table 2: Moment of inertia results measured by CAD.

Axis	DJI MATRICE 100 MOI	Apparatus MOI	DJI MATRICE 100 + Apparatus MOI	Unit
Roll	0.05535	0.05147	0.10682	kgm^2
Pitch	0.05784	0.06739	0.12523	
Yaw	0.10667	0.11854	0.22521	

2 DJI MATRICE 100 MODELING

2.1 Measurement of mass moment of inertia

A prior study of quadrotor modeling estimated the mass moment of inertia assuming a quadrotor's prop and fuselage as cylinders as shown in [4, 5]. However, this method is difficult to obtain accurate mass moment of inertia due to quadrotor geometry and mass distribution. Therefore, in this paper, mass moment of inertia were measured through data postprocessing using CAD design and motion capture camera. For accurate measurement of mass moment of inertia, fixtures were built and CAD models were designed with the same material and size as the DJI MATRICE 100. In addition, for measuring the mass moment of inertia on the roll, pitch, and yaw axis, four motion capture cameras were used to conduct a pendulum motion experiment for two minutes. The DJI MATRICE 100 model information provided by the DJI Corporation is shown in Table 1. Table 2 shows the DJI MATRICE 100 mass moment of inertia measured using a CAD design program. Roll and Pitch axis mass moment of inertia is around $0.05kgm^2$, and the Yaw axis is twice as large as the other axes. As shown in Table 3, because the cross-sectional area of the yaw axis is larger than that of the other axes, the area and weight distribution are larger than those of the other axes. For accurate experiment, DJI MATRICE 100 and fixtures were combined to measure mass moment of inertia. In order to construct the experimental environment using the motion capture camera, as shown in Figure 4, four OptiTrack motion capture cameras were installed at the corners of the cube profile structure and the camera focus was installed $60cm$ from the center of the ground. In order to record the movement of the object using the motion capture camera, the marker was attached to the center of the motor located at the end of each axis of DJI MATRICE 100, and a masking tape was attached to the top surface of the prop due to reflected light during data acquisition. To measure the mass moment of inertia, DJI MATRICE 100 was suspended from the ceiling by connecting KEVLAR yarns of the same length as shown in Figure 2. Experiments were performed for 3 times each for 2 minutes with different lengths of lines. At the same time, data acquisition and image capture of motion capture camera were performed at the same time. The mass moment of inertia is defined as a relational expression as shown in Eqs. 1-3 Figures 5 and 6 show two-dimensional graphs of data acquired for vertical and horizontal axis movements using a motion capture camera. The highest points in the pendulum motion are marked with a red marker to distinguish. Experimental results show that I_{zz} values are close to those calculated by CAD. The roll and pitch axis data using the motion capture camera shows a large error in each experimental case due to the problem of the experimental method. In this study, the mass moments of inertia calculated by CAD were determined to be equal to the model coefficients of the actual DJI MATRICE 100. Also, the roll and pitch axes are the same in consideration of the symmetrical shape of the quadrotor.

$$I_{xx} = \frac{W_1 T_1^2 L_1}{4\pi^2} - \frac{W_2 T_2^2 L_2}{4\pi^2} - \frac{W_3 L_3^2}{g} \tag{1}$$

$$I_{yy} = \frac{W_1 T_1^2 L_1}{4\pi^2} - \frac{W_2 T_2^2 L_2}{4\pi^2} - \frac{W_3 L_3^2}{g} \tag{2}$$

$$I_{zz} = \frac{W_1 T_1^2 A^2}{16\pi^2 L} - \frac{W_2 T_2^2 A^2}{16\pi^2 L} \tag{3}$$

2.2 Aerodynamic database

To construct a dynamics model, an aerodynamic database was constructed using the wind tunnel data of the DJI MATRICE 100 quadrotor performed by KARI. Equation 4 and 5 are the input and output formulas for building the quadrotor

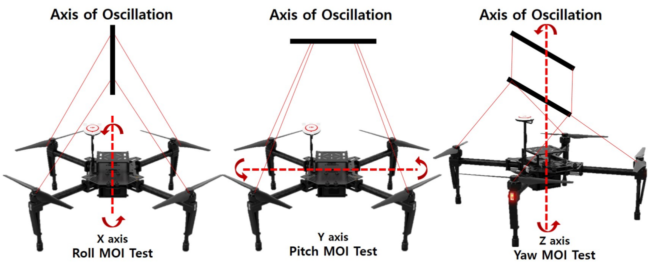


Figure 2: Roll, Pitch, Yaw MOI test axis

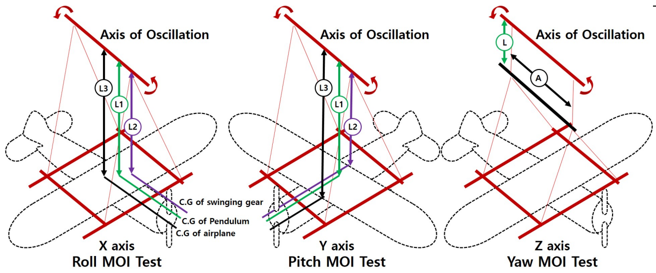


Figure 3: Measurement method of moment of inertia according to flight axis.

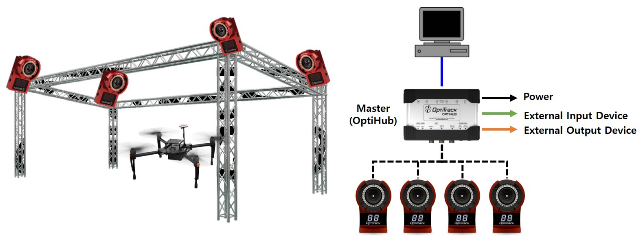


Figure 4: Motion capture camera environment configuration

aerodynamic database. The input variables constitute the Euler angles ϕ, θ, ψ and the body frame velocities U, V, W and RPM.

$$\begin{aligned} X_{body} &= Fn(\phi, \theta, \varphi, U, V, W, RPM, \delta RPM) \\ Y_{body} &= Fn(\phi, \theta, \varphi, U, V, W, RPM, \delta RPM) \\ Z_{body} &= Fn(\phi, \theta, \varphi, U, V, W, RPM, \delta RPM) \end{aligned} \tag{4}$$

$$\begin{aligned} L &= Fn(\phi, \theta, \varphi, U, V, W, RPM, \delta RPM) \\ M &= Fn(\phi, \theta, \varphi, U, V, W, RPM, \delta RPM) \\ N &= Fn(\phi, \theta, \varphi, U, V, W, RPM, \delta RPM) \end{aligned} \tag{5}$$

Because the motor input is different depending on the flight status, the motor RPM must be calculated based on the motor placement of the quadrotor as shown in Figure 7. In Figure 21, the aerodynamic database is constructed for the roll and pitch axes and is divided into blocks for each axis. The force and moment are calculated according to the motor command output. Each block receives the feedback of the state variables of the quadrotor, performs the calculation, and outputs the force and moment.

Table 3: DJI MATRICE 100 Sectional area.

Axis	Sectional area	Unit
Roll (Side)	0.0309	kgm^2
Pitch (Front)	0.0309	
Yaw (Top)	0.0848	

motion.jpg

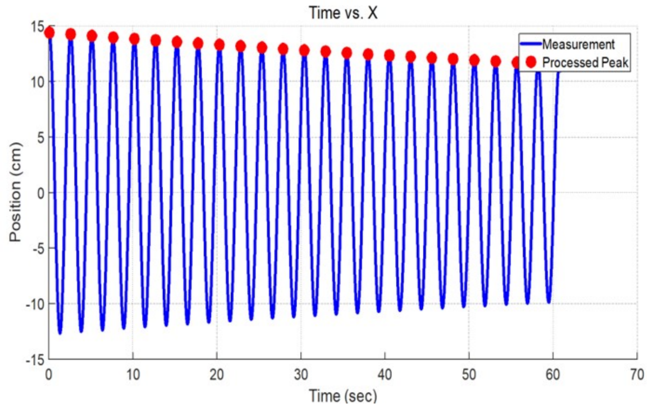


Figure 5: Peak value recording for directional axis pendulum motion

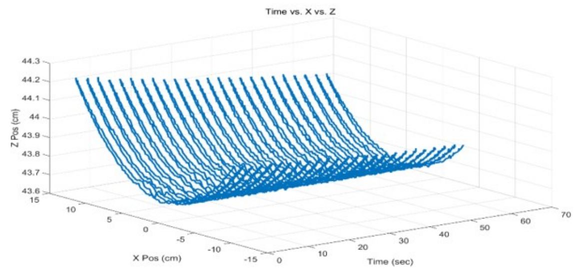


Figure 6: 3-Dimensional graph of transverse pendulum motion

Figure 23 shows that the force and moment are constructed as a multi-dimensional lookup table according to the wind speed. The point outside the reference flight speed is calculated using interpolation. As a result, the aerodynamic database consists of a triple structure and the data for each axis is coupled. Therefore, the output results for each axis must be assigned to equations 6 and 7. Finally, the calculated force and moment are substituted into the quadrotor 6-DOF equation to yield the flight states.

$$\begin{aligned} X_{body} &= X_\theta + Y_\phi \\ Y_{body} &= Y_\theta - X_\phi \\ Z_{body} &= (Z_\theta + Z_\phi)/2 \end{aligned} \tag{6}$$

$$\begin{aligned} L_{all} &= L_\theta + L_\phi \\ M_{all} &= M_\theta - M_\phi \\ N_{all} &= N_\theta + N_\phi \end{aligned} \tag{7}$$

3 Building a simulation environment

3.1 *Quadrotor Dynamics Model*

Figure 8 shows the fixed coordinate system of the quadrotor. We set up the kinematic coordinate system considering

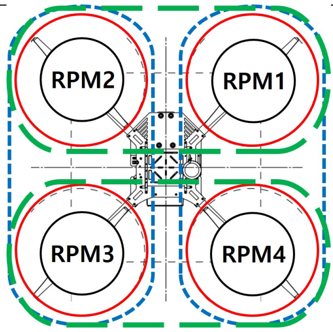


Figure 7: RPM coupling method

Table 4: Average Calculation based on RPM location.

Longitudinal RPM Average Calculation		Unit
Front RPM(1,2)	(RPM1+RPM2)/2	Ω
Rear RPM(3,4)	(RPM3+RPM4)/2	
Lateral RPM Average Calculation		Unit
Right RPM(1,4)	(RPM1+RPM4)/2	Ω
Lest RPM(2,3)	(RPM2+RPM3)/2	

DJI MATRICE 100 based on the X quadrotor frame. The state equations are defined as follows.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} vr - wq + g\sin\theta \\ wp - ur - g\sin\phi\cos\theta \\ uq - vp - g\cos\phi\cos\theta \end{bmatrix} + \begin{bmatrix} X_{body} \\ Y_{body} \\ Z_{body} \end{bmatrix} /m \tag{8}$$

Here, the velocity state equation represents the acceleration on the center of mass of the quadrotor.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L/I_{xx} \\ M/I_{yy} \\ N/I_{zz} \end{bmatrix} - \begin{bmatrix} \frac{(I_{zz}-I_{yy})}{I_{xx}} qr \\ \frac{(I_{xx}-I_{zz})}{I_{yy}} pr \\ \frac{(I_{yy}-I_{xx})}{I_{zz}} pq \end{bmatrix} \tag{9}$$

The angular velocity state equation shows the roll, pitch and yaw changes taking into account the mass moment of inertia, angular velocity and moment generated by the motor and the prop.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta cos\phi \\ 0 & cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{10}$$

Equation 10 is the kinematics between the Euler angular rate and body rate.

3.2 Nominal Flight Controller

In general, the flight controller uses proportional, integral, and derivative control. The attitude controller provides the angle, position and angular velocity of the quadrotor to increase the safety of the quadrotor system by the P-PID structure. The altitude controller is based on a PID controller. The position controller is based on a PD controller.

$$\begin{aligned} \ddot{x} &= \{U_1(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) - K_1\dot{x}\}/m \\ \ddot{y} &= \{U_1(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) - K_2\dot{y}\}/m \\ \ddot{z} &= \{U_1(\cos\phi\cos\psi) - K_3\dot{z}\}/m - g \end{aligned} \tag{11}$$

In equation 11, K_1, K_2 and K_3 are the proportional gain in the attitude control loop.

$$\begin{aligned} u_1 &= \delta_{thr} = (+\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ u_2 &= \delta_{ail} = (-\Omega_1^2 + \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\ u_3 &= \delta_{ele} = (-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ u_4 &= \delta_{rud} = (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{aligned} \tag{12}$$

The control input is shown in equation 12 and means the RPM command for each motor. $\delta_{thr}, \delta_{ail}, \delta_{ele}$ and δ_{rud} are the virtual control input of the quadrotor by mixing the command values from the controller in Equation 12. u_1, u_2, u_3 and u_4 are the RPM.

Based on the X-shaped quadrotor shape as shown in Figure 8, the command is transmitted to the motor through the control mixing by the roll, pitch, yaw and altitude commands.

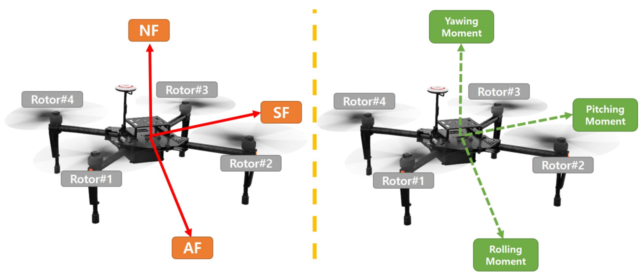


Figure 8: Quadrotor frame of reference

3.3 Disturbance Observer Based Control

Many controllers are designed based on the mathematical modeling of the system to be controlled [6, 7, 8]. Most controller are designed assuming no disturbance or are designed with worst-case assumptions. If disturbance exists, there is a possibility that the system becomes unstable because the nominal performance cannot be maintained and follow-up error occurs. There is a disadvantage in sacrificing desired control performance such as fast and accurate tracking performance in the absence of disturbance. However, most real-world systems do not fit mathematical modeling, and additional disturbances that are not taken into account from external environments are often applied to the system. These uncertainties can degrade controller performance and make the system unstable. In this paper, we propose a robust control method that stabilizes the system and guarantees the control performance even in the presence of model uncertainty and disturbance. The DOBC (Disturbance Observer Based Control) was designed by combining with the PID controller. The DOBC is a technique for observing or estimating the effect of model uncertainty and disturbance on the system. It can be estimated by filtering data with uncertainties such as disturbance due to an external load of the motor, frictional force, gust and incomplete dynamics. The DOBC can compensate for these uncertainties. In addition, it can be designed by patching on the existing controller, and it is possible to guarantee the performance of the existing controller by compensating the disturbance. The DOBC depends on the design of the Q-filter, which is the nominal dynamics that the designer desires to model an uncertain system. There is a nominal performance restoration characteristic when there is no disturbance by deriving an additional control input in case of disturbance. And the controller technique provides robustness and adaptability. The DOBC block is shown in Figure 9. $C(s)$ has a basic PID structure as shown in equation 13.

$$C(s) = K_P + \frac{K_I}{s} + \frac{K_D s}{T_f s + 1} \tag{13}$$

Here, the PID controller is created through MATLAB Simulink, and T_f is set to 1 at this study. The disturbance observer is the key to the estimation input given to the real model and the inverse model. Knowing the exact $P(s)$ is difficult, the nominal model $P_n(s)$ can be obtained. Therefore, if the output of the model and the control input are given, we estimate the disturbance so that the DOBC compensates it. The Q-filters filters the noise and makes a proper transfer function of $P_n(s)^{-1}$. The control input shown in the DOBC structure can be expressed as in equation 14

$$u = \bar{u} + (y_P - \hat{u}_p) = \bar{u} - \hat{d} \tag{14}$$

$$u = \bar{u} + u_{filtered} - \frac{1}{J_{xx}}\dot{x}_{2.filtered} = \bar{u} + y_p - \hat{u}_p \tag{15}$$

Where $\bar{u}$ is a nominal control input from the PID controller. The difference between the filter control input y_2 and the control input $\hat{u}_2$, which includes the effect of the disturbance estimated through the inverse nominal model. Nominal model equation and attitude transfer function are represented in equations 16 - 17. Equation 18 shows the nominal model transfer function about roll control.

$$L = J_{xx}\ddot{\phi} \tag{16}$$

$$L = c_t \delta_{ail} - (J_{zz} - J_{yy})\dot{\theta}\dot{\psi} \tag{17}$$

$$P_n(s) = \frac{\phi(s)}{\delta_{ail}(s)} = \frac{1}{J_{xx}s^2} \tag{18}$$

The Q-filter plays an important role in the DOBC for robust state and estimate disturbance. Q-filter should be designed to have at least the same relative order of nominal dynamics as shown in Equation 19.

$$Q_A(s) = Q_B(s) = \frac{a_0/\tau^2}{s^2 + (a_1/\tau)s + (a_0/\tau^2)} \tag{19}$$

Among the filter parameters, λ determines the accuracy of the disturbance estimation as an adjustable parameter. The smaller the λ, the better the transient response state, but the system may become unstable. The advantage of the DOBC is that it works only in disturbance situations and provides active anti-disturbance control with adaptability and robustness.

4 Numerical Simulation

Comparing the case with and without the DOBC, the attitude and position tracking performance were analyzed by calculating RMSE (Root Mean Square Error).

$$\begin{aligned} \dot{p}_{dist} &= 7\cos(t)\ (rad/s^2) \\ \dot{q}_{dist} &= 7\cos(t)\ (rad/s^2) \\ \dot{r}_{dist} &= 5\cos(t)\ (rad/s^2) \end{aligned} \tag{20}$$

In Equation 20, the disturbance is applied to the pitch, roll, and yaw axes and multiplied by the mass moment of inertia to affect the moment equation. Figure 10 shows the simulation result using the PID controller. The X position cannot follow the position command after 40 seconds, and the Y position does not follow the command for the entire time domain. Figure 11 shows the result of the thrust command and

RPM using the PID controller. The thrust command shows slow response and the noise mixed in command.

Figure 12 shows the result of the simulation performed by combining the PID controller and the DOBC. Unlike the results by the PID controller, it can be seen that it follows the attitude and position command well. As shown in Figure 13, the RPM control also compensates for the disturbance estimated by the DOBC, while the noise is also reduced compared with the PID control while applying appropriate thrust command.

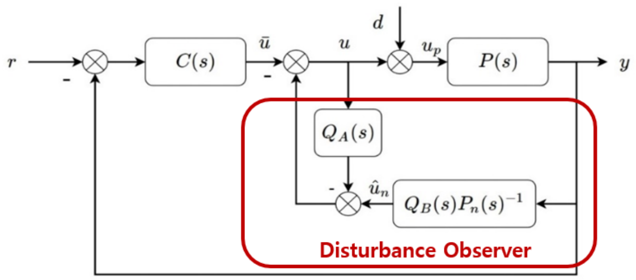


Figure 9: DOBC configuration

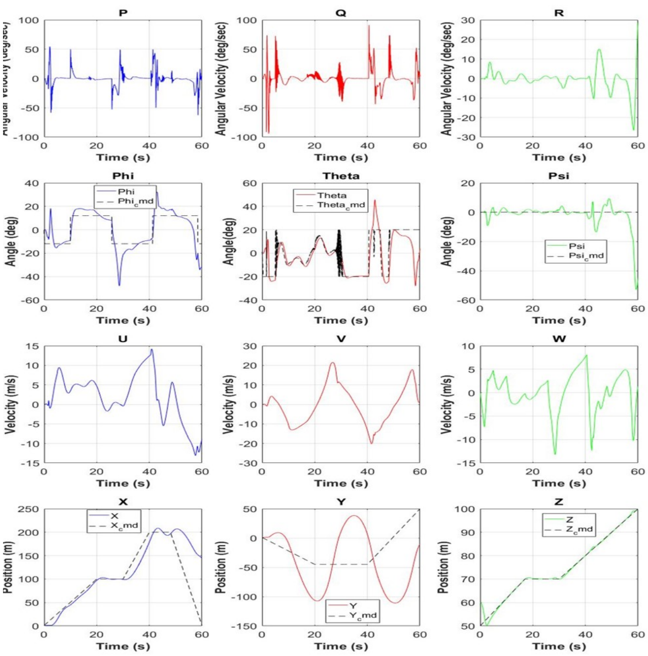


Figure 10: Simulation result - states (PID Controller).

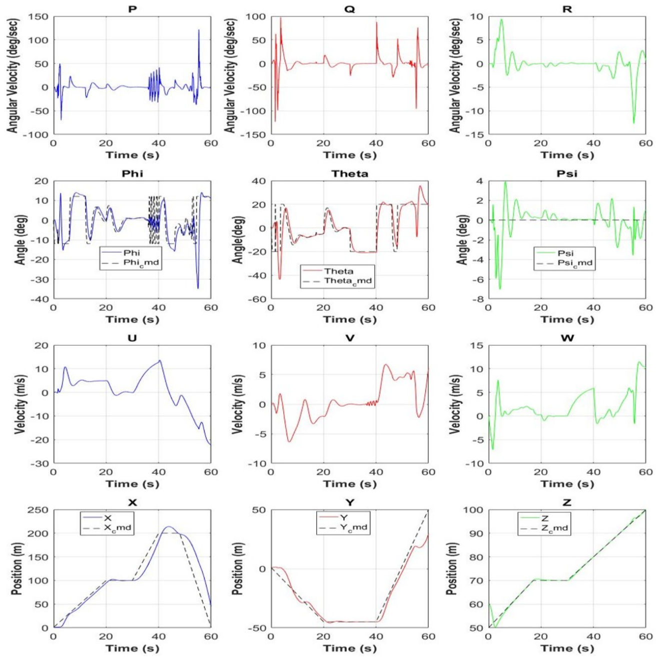


Figure 12: Simulation result - states (DOBC Controller).

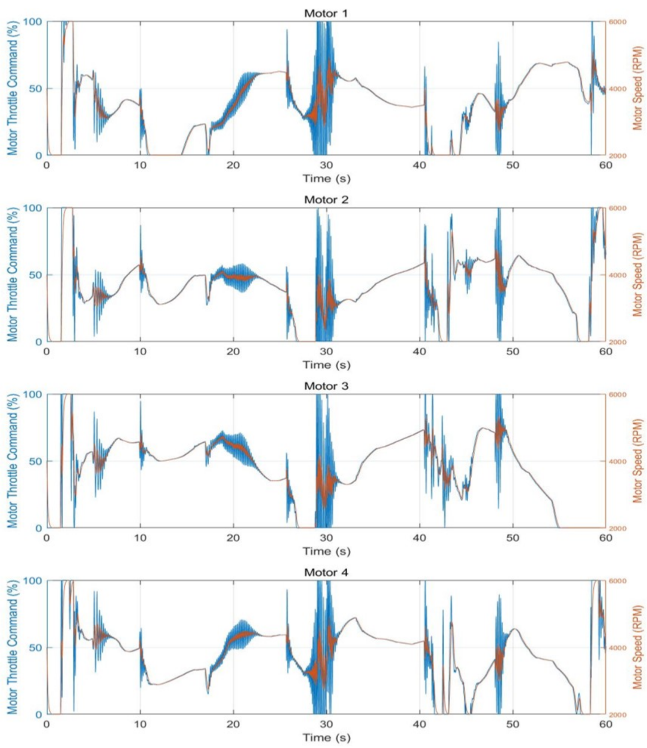


Figure 11: Simulation result - RPM (PID Controller).

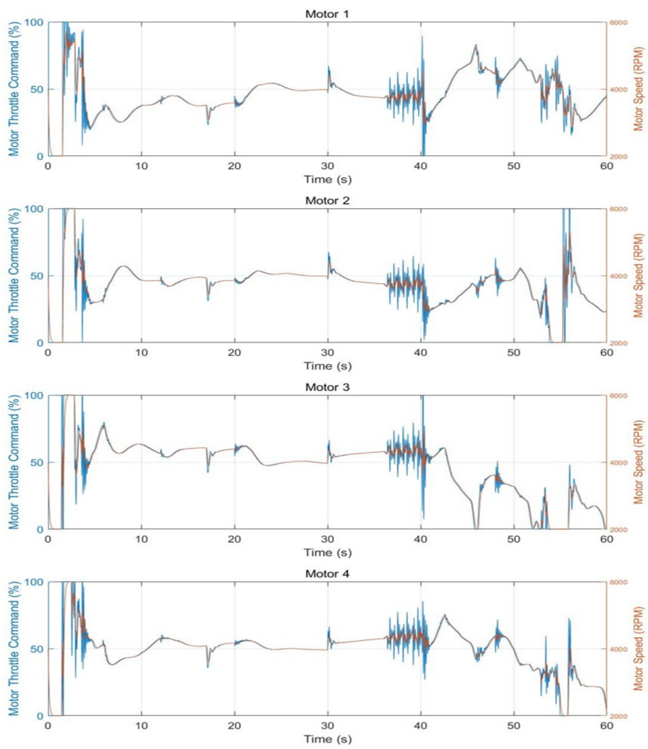


Figure 13: Simulation result - RPM (DOBC Controller).

Table 5: Simulation RMSE results comparison.

RMSE result (PID)		RMSE result (DOB)	
Phi (Degree)	0.1676	Phi (Degree)	0.1290
Theta (Degree)	0.1920	Theta (Degree)	0.1624
Psi (Degree)	0.2827	Psi (Degree)	0.1493
X (Position)	42.2538	X (Position)	24.2384
Y (Position)	64.4534	Y (Position)	6.2603
Z (Position)	1.3536	Z (Position)	1.3455

In order to numerically analyze the simulation results of the two control methods, the RMSE was calculated based on the simulation results. From the RMSE results, it can be seen that the simulation results using the DOBC against the attitude and position commands are improved than those using only the PID controller. The main tuning parameters of the DOBC were obtained numerically. The flight test was performed based on the DOBC variables obtained from the simulation.

5 Flight Test

5.1 Manufactured Quadrotor

The UASG-DOBC-Quadrotor shown in Figure 14 was developed so that aerodynamic modeling based on the aerodynamic data measured through the open wind tunnel experiment and aerodynamic characteristics is similar to the actual quadrotor. The FCC (Flight Control Computer) equipped in the UASG-DOBC-Quadrotor is Pixhawk Cube and the firmware is Ardupilot. The DOBC was combined with the PID - based attitude controller implemented in Ardupilot. We implemented the function to enable the DOBC according to the pilot command on the ground. The main control parameters of the DOBC were remotely modifiable in GCS (Ground Control System).

5.2 Performance Measurement Experiment of a Blower Fan

A blower fan produces disturbance environment. When flight tests are performed indoors, GPS accuracy drops. Therefore, the experimental environment was constructed as shown in Figure 15. The blower fan performance test was carried out using the climate measurement equipment to confirm the exact wind speed performance according to the distance. The measurement results are shown in Figure 16. The maximum wind speed was measured to be $14.6m/s$ and the wind speed decreased at a constant interval as the distance increased. In this study, the quadrotor will enter the disturbance path at a distance of about 2m from the blower fan, and then the attitude and position response of the quadrotor will be checked according to the control method.

5.3 Flight Test

In order to verify the performance of the DOBC before the waypoint flight test, a control performance experiment was performed using a blower fan. The test method is shown in Figure 17. In order to simulate the disturbance environment, the horizontal and vertical winds were implemented. The design variables of the DOBC were tuned through this test. Among them, when the τ is greatly reduced, the quadrotor itself oscillates even though it is not affected by the disturbance. After tuning the DOBC, we carried out the flight test using the dummy weights and finally performed the waypoint flight. As shown in Figure 19, the flight test using the dummy weight was carried out in three ways. Figures 24-27 show the flight test results for Method 3 among the flight tests shown in Figure 19. The flight test was conducted by disturbing the quadrotor by a person on the ground when hovering at an altitude of 4 meters. In order to have a periodic disturbance in the quadrotor, a person on the ground periodically pulls the string strongly. Figure 26 and 27 show that the DOBC as a whole greatly reduces the noise and follows the command value. Here, we can see the structural limit of the quadrotor. It can be seen that the yaw axis control force is insufficient due to the disturbance generated on the roll and pitch axes. Therefore, it is necessary to increase the control force on the yaw axis by rotating the motor thrust vector direction of the quadrotor in a future study. Figures 28 are the flight path results from the waypoint flight. As shown in Figures 19, the

Figure 14: UASG-DOBC-Quadrotor.

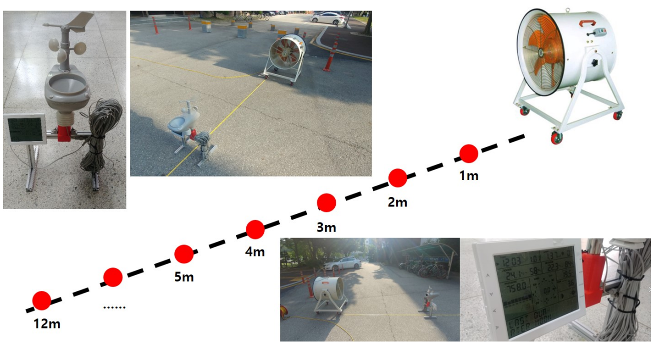


Figure 15: Performance measurement experiment of a portable blower fan.

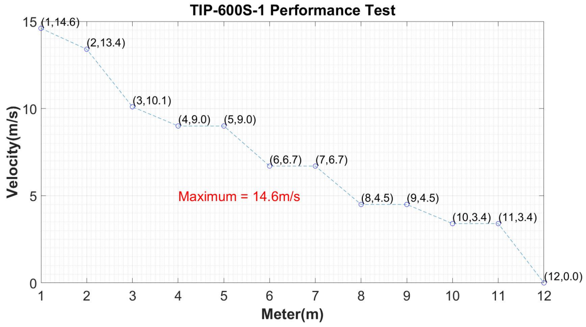


Figure 16: Blower fan performance test.

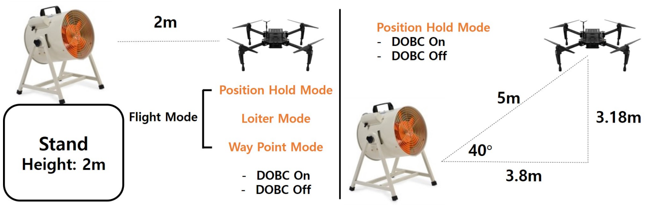


Figure 17: Flight test method using blower fan.

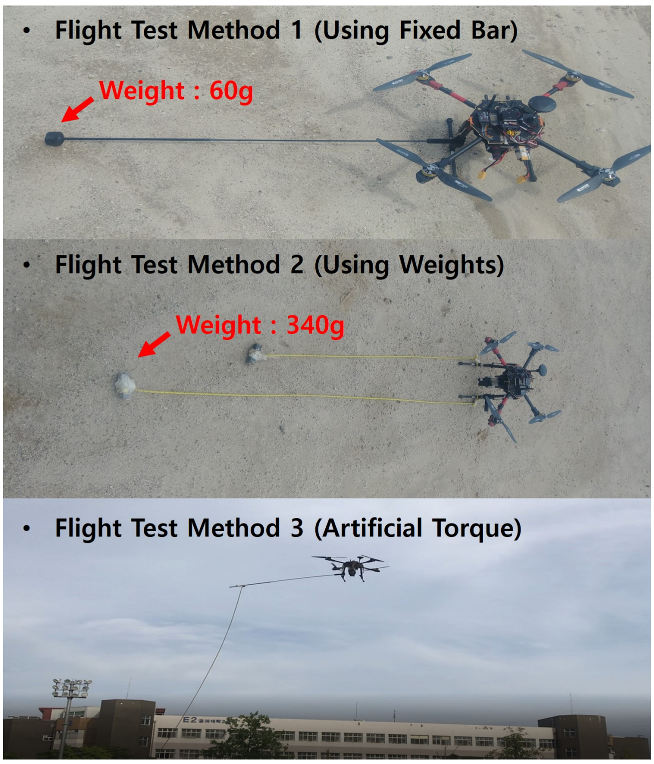


Figure 18: DOBC performance verification flight test method.

Table 6: Flight test RMSE results comparison

Contents	RMSE result (PID)	RMSE result (DOB)
Method 3 (Using Human Force) RMSE		
Phi rate (deg/sec)	0.3137	0.2570
Theta rate (deg/sec)	0.0659	0.0681
Yaw rate (deg/sec)	0.3076	0.3210
Phi (deg)	0.4245	0.2772
Theta (deg)	0.1013	0.1067
Yaw (deg)	0.5534	0.5032
Waypoint Flight Test RMSE		
Phi rate (deg/sec)	0.1069	0.1032
Theta rate (deg/sec)	0.1163	0.1094
Yaw rate (deg/sec)	0.3472	0.4499
Phi (deg)	0.1977	0.1882
Theta (deg)	0.2705	0.2408
Yaw (deg)	3.6156	3.5352

waypoint flight test was performed by attaching the weight to the quadrotor landing gear as shown in Method 2.

As the quadrotor moved, the weight tied to the line caused a moment in the quadrotor by pendulum movement in an unexpected direction. It can be seen that the DOBC follows the similar flight path by applying the DOBC even if the weight disturbance occurs. The RMSE (Root Mean Square Error) results for flight tests are shown in Table 6. From the angular velocity and Euler angle results, it can be seen that the error is smaller when the DOBC is applied. However, the lack of control yaw axis can be seen in waypoint flight test case. Therefore, future studies will improve this problem and proceed with the waypoint flight test.

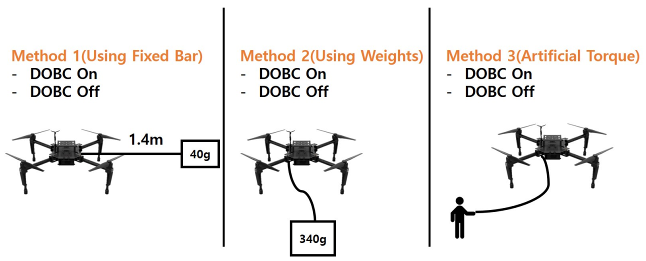


Figure 19: Flight test method with slung load uncertainty.

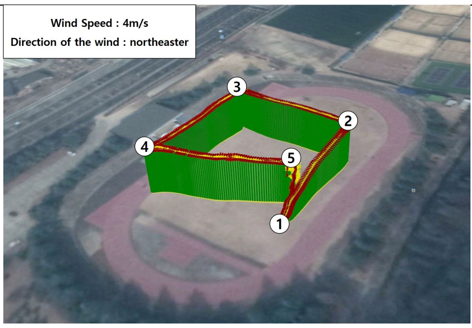


Figure 20: Waypoint flight path.

6 Conclusion

Simulation and flight tests manifested that the PID controller in disturbance environment has limitations and is impossible to keep robust control. In addition, when the DOBC was applied, it was possible to control the airframe by compensating the disturbance with uncertainty. The DOBC showed better performance in a strong disturbance environment. Such as slung load, artificial wind, and human interaction. Thus, the DOBC could effectively compensate the disturbance and increase flight stability of the quadrotor system. Future work DOBC will be done by patching on the position controller. And we propose a collision avoidance algorithm using DOBC using relative navigation and collision avoidance algorithm in a disturbance environment implemented in a room.

Acknowledgements

This work is supported by the research project (No. 10062497) funded by the Ministry of Trade, Industry and Energy of Korea, Republic of.

References

[1] S. Bouabdallah and R. Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *IEEE International Conference on Robotics and Automation*, 2005.

[2] M. Bouchoucha H. Bouadi and M. Tadjine. Sliding mode control based on backstepping approach for an uav type-quadrotor. *World Academy of Science Engineering and Technology*, 26(5):22–27, 2007.

[3] A. M. Ashfaq and D. Wang. Modeling and backstepping-based nonlinear control strategy for a 6 dof quadrotor helicopter. *Chinese Journal of Aeronautics*, 21(3):261–268, 2008.

[4] J. J. Lee. Experimental determination of mass moments of inertia for tilt rotor uav (unmanned aerial vehicle). In *The Korean Society for Aeronautical and Space Sciences Spring Conference*, 2013.

[5] M. P. Miller. An accurate method of measuring the moments of inertia. In *NASA Reports, U.S., Report*, 1996.

[6] S. K. Kim S. H. Jo, H. J. Jeaong and J. Y. Suk. Robust control of the small coaxial compound helicopter uav using disturbance observer based control. In *2017 Asia-Pacific International Symposium on Aerospace Technology*, 2017.

[7] W. Ha K. Lee and J. Back. Overview of disturbance observation techniques for linear and nonlinear systems. *Journal of Institute of Control, Robotics and Systems (in Korean)*, 22(5):332–338, 2016.

[8] S. H. Jeong and S. Jung. Design and experimental studies of a disturbance observer for attitude control of a quadrotor system. *Journal of Institute of Control, Robotics and Systems (in Korean)*, 19(11):1004–1010, 2013.

APPENDIX A: AERODYNAMIC MULTIDIMENSIONAL DATABASE

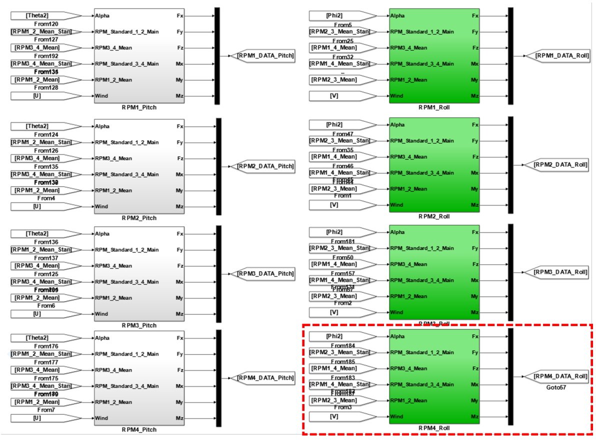


Figure 21: Aerodynamic database according to RPM number

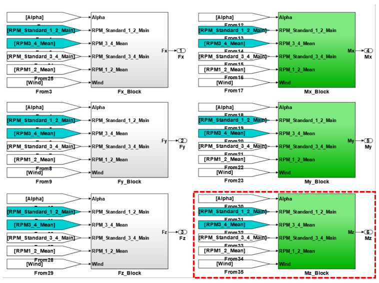


Figure 22: Force and moment aerodynamic database

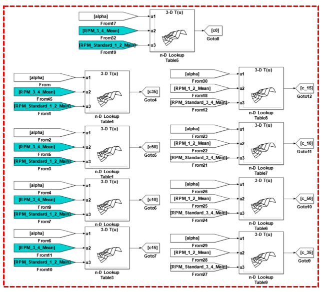


Figure 23: Aerodynamic database based on speed

APPENDIX B: FLIGHT TEST DATA

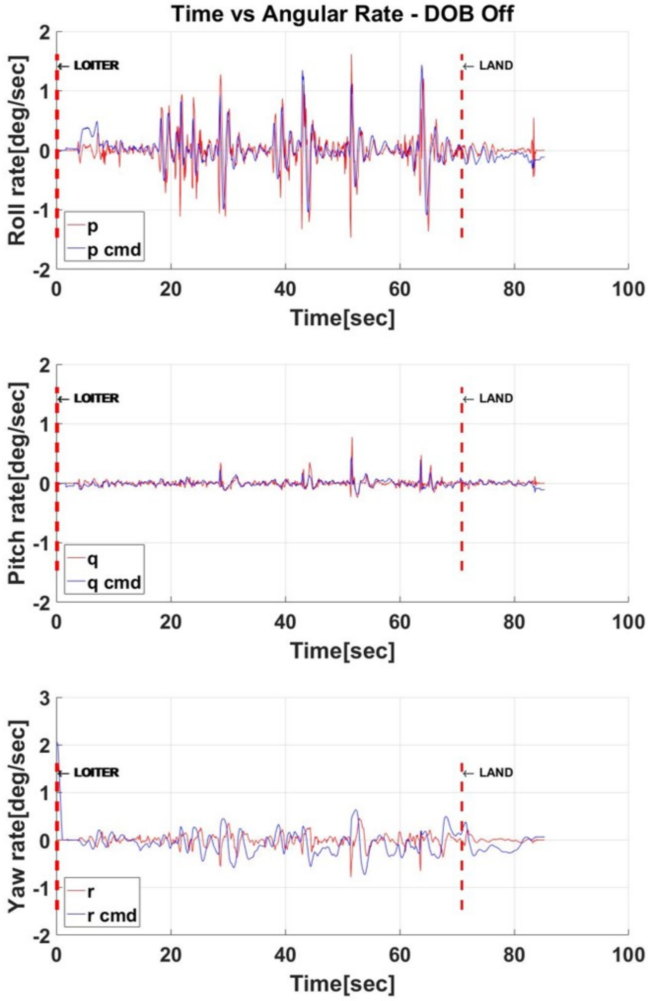


Figure 24: Method 3: Angular rate DOBC OFF.

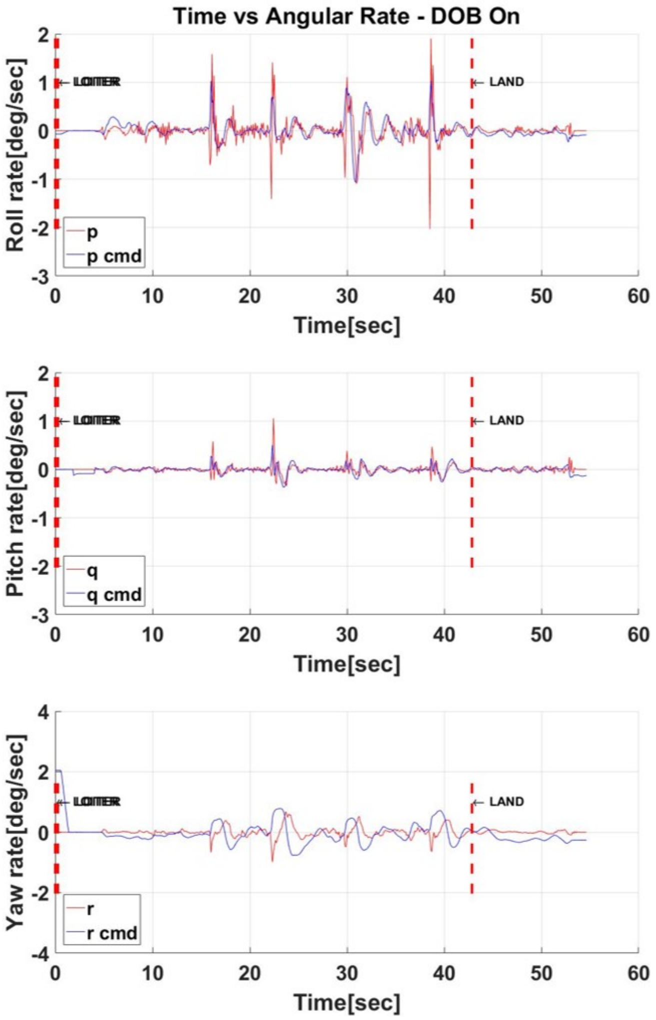


Figure 25: Method 3: Angular rate DOBC ON.

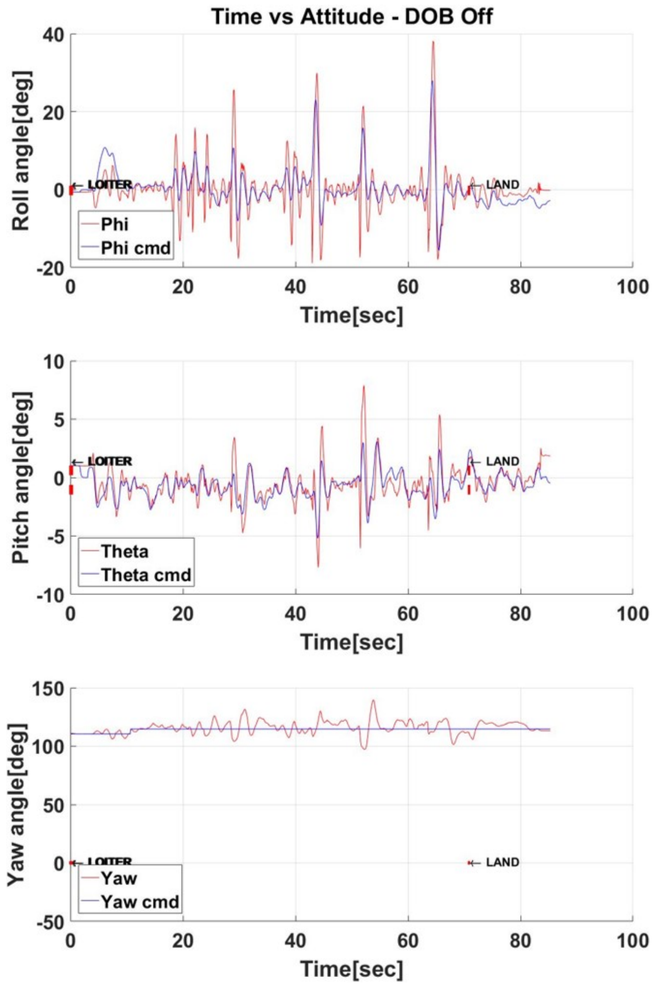


Figure 26: Method 3: Attitude DOBC OFF.

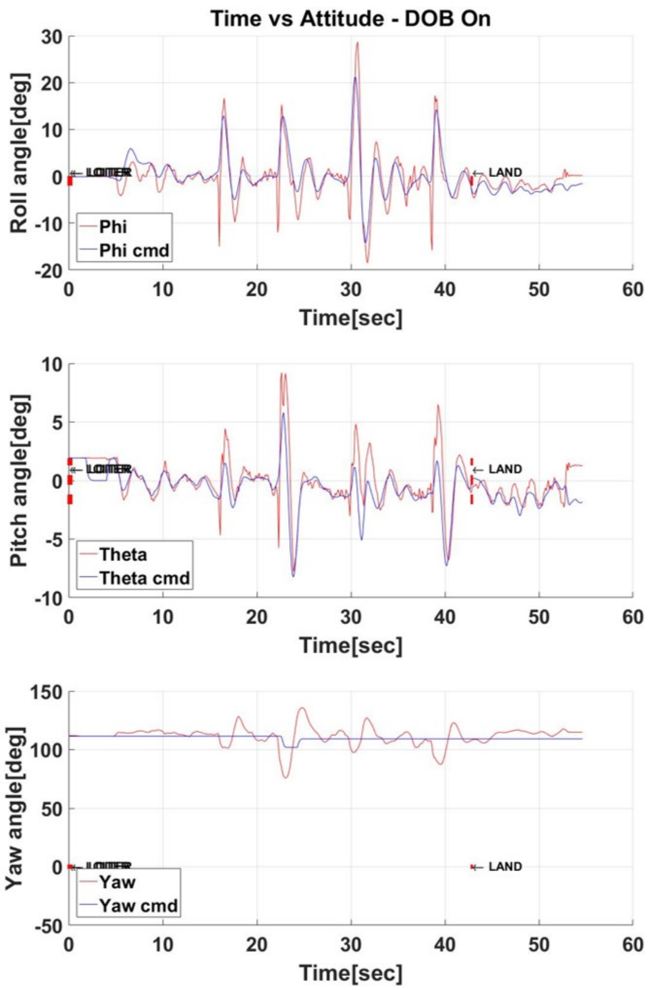


Figure 27: Method 3: Attitude DOBC ON.

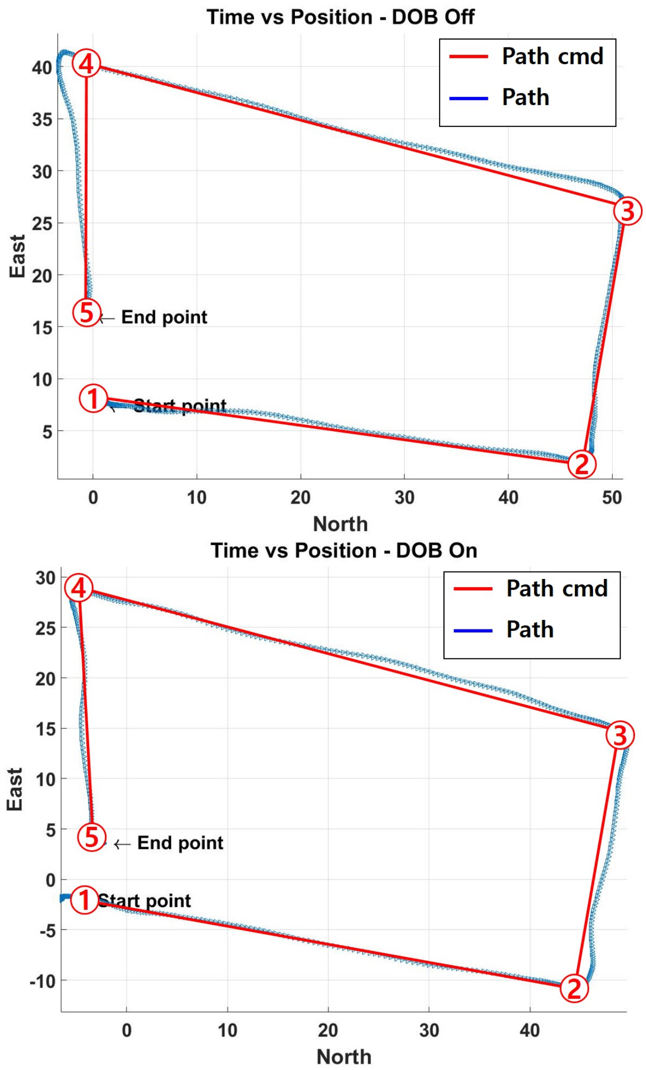


Figure 28: Waypoint flight: Path DOBC OFF / ON.

Influence of the Target Material and Disturbance Sources in the Accuracy of Distance Sensors for MAV Applications

R. L. S. M. Vilela [1,2] *, H. P. Saraiva [1,2] †, E. S. P. Lopes [1,2] ‡, R. C. M. Pires [1,2] § and E. C. Silva [1,2] ¶

[1] Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro, PUC-Rio, Rio de Janeiro, Brazil

[2] AeroRio, Pontifical Catholic University of Rio de Janeiro, PUC-Rio, Rio de Janeiro, Brazil

ABSTRACT

Distance sensors are widely used in an extensive variety of applications nowadays, including obstacle detection and avoidance for autonomous navigation. There are multiple options, based on different working principles, possessing different distance ranges, reading rates and accuracies that could vary according to the type of sensor and the environment of operation. This paper focus in analyzing the response of distance sensors for UAV applications. The chosen sensing technologies are ultrasonic, infrared and LiDAR, which were selected based on several parameters, such as, weight and accuracy. The sensors response was evaluated considering different target materials (wood, cardboard, polyethylene foam and Perspex), aiming to analyze their behaviour towards each material. Initially, in a controlled environment, without disturbances, a target was placed at several known distances from the analyzed sensor, within its operational range. For each one of these distances, the sensor output was compared with the known distance, in order to evaluate its accuracy. This procedure was repeated for all of the analyzed sensors. Furthermore, aiming to verify how much the sensors performance deteriorated in the presence of disturbance sources, the same procedure was repeated to the sensors em- bedded in a MAV, with its motors and propellers running during the test, aiming to replicate real flight conditions. The obtained results lead to conclusions about which kind of technology is more appropriate for this application, providing more reliable measurements, with increased accuracy, and, consequently, allowing the performance enhancement of collision avoidance algorithms.

*Email address: renandelima95@gmail.com

†Email address: saraiva.henriquep@gmail.com

‡Email address: lopespserico@gmail.com

§Email address: rodrigocamargo95@gmail.com

¶Email address: edusilva@ele.puc-rio.br

1 INTRODUCTION

Nowadays, Unmanned Aerial Vehicles (UAVs) are being employed in an increasing number of fields, such as environmental monitoring, border patrol, search and rescue operations, disaster relief, among others. Besides, in the next years, UAVs market is expected to provide billions of dollars in economic growth since it is rapidly growing in a lot of civilian and commercial industries such as agriculture, energy, utilities, mining, construction, real estate, news media and film production [1]. These applications often require small and agile UAVs, capable to fly at low altitudes or even inside buildings, becoming exposed to many hazards and obstacles. However, current UAVs technology in automatically sensing, detecting and avoiding fixed and moving obstacles is still immature compared to manned aerial vehicles. Obstacle detection is essential for collision avoidance systems, that play a key role in autonomous navigation. Several types of sensors can be placed in a UAV to detect and identify obstacles along its path. The data acquired by these sensors are gathered and processed using collision avoidance algorithms, that define the avoidance action based on the processed obstacles information [2].

It was chosen to analyze sensors that measure distance. The examined technologies are ultrasonic, infrared and LiDAR. These three types are vastly used in drone applications, hence, justifies deeper investigation. Ultrasonic sensors measurements are made via sound wave propagation, so, noisy environments might interfere with the results accuracy if there is a sound component in the same frequency of the wave emitted by the sensor [3, 4] . Infrared sensors, on the other hand, emits an infrared beam and measures the intensity of its reflection. Consequently, for targets with too high or too low reflectivity the measurements are not reliable. Another kind of environment that harms the accuracy of infrared sensors are those which have much infrared light emission, however, these are not very common. LiDAR functions by emitting a laser pulse and measuring the phase difference between the emitted and received wave. Therefore environments with high power wave propagation in the LiDAR's frequency might interfere in its functionality.

Once the sensor process is known, it is possible to analyze how the measurement is affected by the reflection. Different materials produce different reflection distortions, thus, it is also important to characterize how they interfere on the final result. The materials chosen to examine are wood, cardboard, polyethylene foam and Perspex. This selection is due them being relatively common in collision avoidance problems. Furthermore each one has diverse structural properties which make the analysis more complete.

Usually, knowing the right sensing technology for an application is not an easy task, since it requires to study several variables included in the system and his surroundings.To simplify this study, the sensors were tested in an minimum noise environment, with and without an UAV system, i.e, with and without being embedded in an MAV. The objective is making minimal the dependency of the location and maximize the correlation between the material, system properties and the sensors' measurements [5].

This paper proposed experiment should be useful for precising the reliability of the sensors considering only inner MAV disturbances and the effect caused by different materials. Provided by this results, the decision of which sensor should be used for each case tends to be more precise because it gives a relation of real sensor accuracy considering the materials of environment's application.

2 Sensors

As described in section 1, nowadays the use of autonomous drones is becoming a trend in the modern society. There are many ways to implement an autonomous drones, but there are some basic sensing that are required to make it possible. So it's reasonable to assume that some sensors cannot be removed completely, especially distance sensors. This study analyses the response of 3 sensors that are mostly used for that purpose but with different working principles. In Sections 2.1, 2.2 and 2.3, a brief presentation about the characteristics and specifications of the sensors used is made.

2.1 Ultrasonic

The ultrasonic sensor tested was the MaxBotix MB1242, which sensor is fully equipped with a real-time noise rejection and real-time auto calibration features. This sensor has a internal circuit that allows a very easy-to-use experience. It converts the reading from the sensor into a 16 bit integer that is split into two 8 bit numbers there is sent through the I^2C bus. The 16 bit number is the range that the sensor read in millimeter. There is only one easy and well documented I^2C address and mode configuration there has to be done so that the sensor can start sensing.

The ultrasonic sensor has a range of 20 cm to 765 cm. It has also a digital output of 16 bit for the reading, so the resolution is fixed and has a precision of 1 cm, it has a 40 Hz acquisition rate, only weights 6 g and a angle of acceptance of 19^o.

2.2 LiDAR

The LiDAR sensor tested was a Makerfocus LiDAR range finder (TFmini Infrared Module). It has, like the ultrasonic sensor, a circuitry designed to convert the reading from the sensor to a serial output. The data from that reading is, also, a 16 bit number that is split into two 2 bytes number, sent trough the serial output and read in a micro-controller.

This sensor weights 6.1 g, has a 100 Hz acquisition rate, a 2.3^o acceptance angle, an accuracy of 1% at any distance lesser than 6 m, the precision is 2 cm and a range of 30 cm to 1200 cm.

2.3 Infrared

The infrared sensor tested was a Sharp GP2Y0A02YK0F IR Range sensor. Unlike the others sensor tested, the response of it is not linear nor easy to use right out of the box and can be seen in Figure 1. So to make it more easy to use, a function fitting was needed. This can be done in many different ways, one of them is using the points shown in the response of the sensor , using Excel's fitting function (knowing that the response of the sensor follows the diode law) and them trying to solve the fitted function in voltage (or y) instead of distance (or x).

$$y = 10650.08 \times x^{-0.935} - 10 \quad (1)$$

This infrared sensor has a range of 20 cm to 150 cm with analog output that varies from 2.8 V (at 20) to 0.4 (at 150 cm) V, so the resolution depends only on the ADC – for the rig used the resolution was 4.88 mV), with 2 cm precision, an angle of acceptance of 5^o, the acquisition rate can be limited to approximately 20.8 Hz to 35.7 Hz. And weights only 5 g, the lightest of them all.

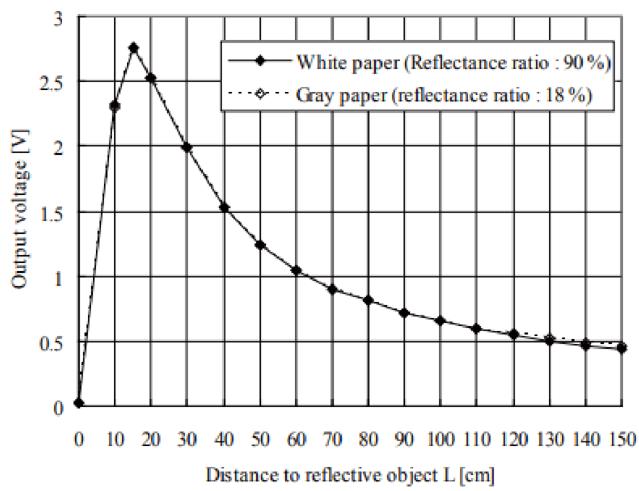


Figure 1: Output of the sensor vs. the distance.

To make it simple and direct, Table 1the summarizes the sensor's specifications and characteristics.

Specification	Ultrasonic	LiDAR	Infrared
Precision [cm]	1	2	2
Range [cm]	20 - 765	30 - 1200	20 - 150
Angle []	19	2.3	5
Rate [Hz]	40	100	26.3
Weight [g]	6	6.1	5

Table 1: Sensors specification summary

3 METHODOLOGY

The measurements were performed using a class 250 quad-copter MAV. The MAV‘s powertrain is specified in Table 2. Its control is done with the Navio2 autopilot platform, connected to a Raspberry Pi 3 running the ArduPilot firmware and radio controlled using a RFD900x telemetry radio with PPM pass-through.

Component	Specification
Motor	E-Max 1806 - 2280
Propeller	Carbon Fiber - 3-Blade - 5X3.5
ESC	E-max BLHELI-S 12A 2-4S
Battery	LiPo 3S (11.1 V) - 1400mAh - 40C

Table 2: Powertrain specifications.

The sensors were mounted individually in the MAV (i.e. during the infrared test, only the infrared sensor was mounted in the MAV) due to sizing limitations and in order to remove any chance of interference of different sensors technologies. They were positioned in the MAV's direction of movement in a place that the propellers would not be inside its field of view.

Figure 2 shows a picture of the MAV used with the LiDAR sensor mounted, right below the camera.

Figure 2: MAV used during the tests with LiDAR sensor mounted.

For the ultrasonic and LiDAR sensors, the measurements were taken within a range from 0 to 300 cm, increasing the distance in steps of 10 cm. For the infrared sensor, the range used was from 0 to 150 cm, with steps of 10 cm. This difference in the range for the IR sensor is due to its nominal range being smaller than the other sensors.

The target material dimensions were selected based on the sensor's field of view to guarantee that the sensor would be measuring only the intended material. The dimensions were calculated considering the biggest distance that was going to be measured (300 cm). Table 3 shows the field of view radius for the three sensors considering the distance of 300 cm.

Sensor	FOV Angle	FOV Radius @ 300 cm
Ultrasonic	19^o	50.2 cm
LiDAR	2.3^o	6.02 cm
Infrared	5^o	13.1 cm

Table 3: Sensor's field of view.

All the target material were selected with dimensions bigger than 100.4 cm. The MAV was positioned at a height that would permit the sensor to detect only the target material, without being influenced by the floor. The setup of the test is represented in Figure 3.

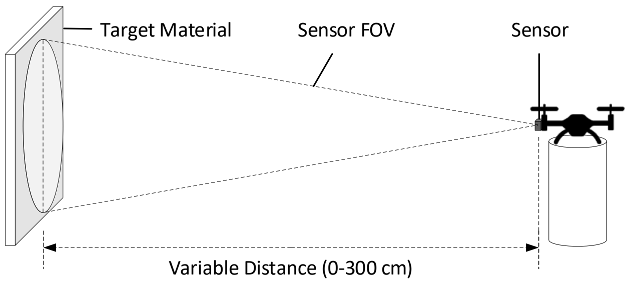


Figure 3: Test setup

To analyze the sensor‘s performance for different target materials, and to provide a benchmark comparison for the following test, the first round of tests were performed with the MAV motors powered off, that can be considered an scenario in which the sensors would provide the best performance. This controlled environment permits to decouple the analysis of the varying performance according to the different target materials to the analysis of the influence of the disturbance sources.

To simulate real flight conditions, considering vibration from the motors and turbulence caused by the propellers, the same test were performed with the motors powered to 70% of throttle while the MAV was attached to a 10 kg steel block that did not permit its movement. This restriction of movement allowed the correct positioning of the sensor to permit accurate comparisons with the measured distance.

The procedure for both the tests was strictly the same, the measurements started from the closest distance (10 cm) and it was increased in steps of 10 cm until it reached the maximum distance (150 cm to the infrared sensor and 300 cm to the ultrasonic and LiDAR sensors). Then, the distance started decreasing to provide a more accurate representation of the measurement and to reduce the sensor's hysteresis influence on the result. The values shown in Section 4 are the average between the increasing and the decreasing measurement. Since this paper do not take in consideration timing response of the sensors, a waiting period of 5 seconds was performed between measurements to allow proper stabilization of the measurement.

In total, 24 tests were conducted and the results are shown in Section 4.

4 Results

The experiment described in Section 3 was performed for the three sensors and the Sections 4.1, 4.2 and 4.3 shows the results obtained. After compiling all the acquired data from the sensors, a statistical analysis was performed in order to evaluate the results.

4.1 LiDAR Results

Figures 4 and 5 show the distances measured by the LiDAR sensor in function of the actual distance, for different target materials: wood, cardboard, polyethylene foam and Perspex. The measurements presented in Figure 4 were performed with the MAV motors turned off, while the results presented in Figure 5 were acquired with the MAV motors turned on.

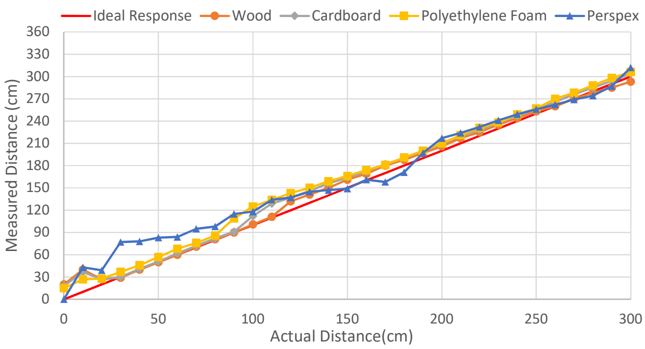


Figure 4: LiDAR experimental response with electric motors off, for different target materials

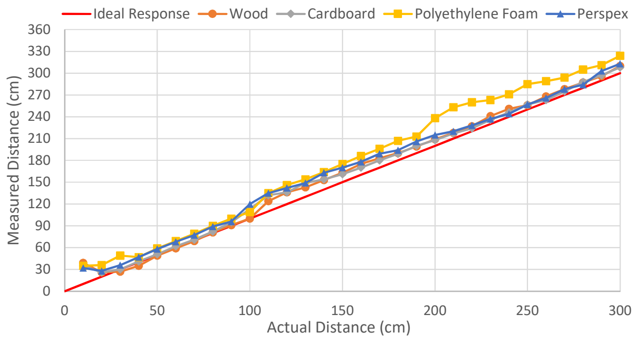


Figure 5: LiDAR experimental response with electric motors on, for different target materials

4.2 Ultrasonic Results

Figures 6 and 7 show the distances measured by the ultrasonic sensor sensor in function of the actual distance, for different target materials: wood, cardboard, polyethylene foam and Perspex. The measurements presented in Figure 6 were performed with the MAV motors turned off, while the results presented in Figure 7 were acquired with the MAV motors turned on.

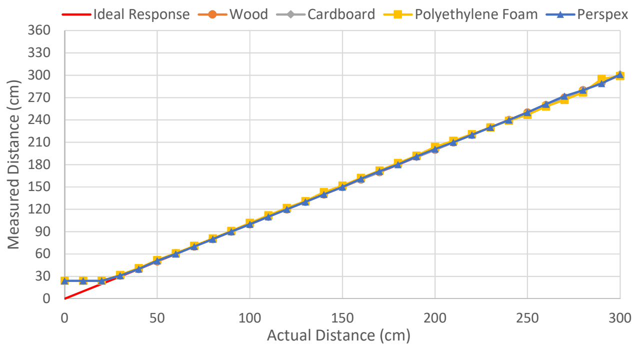


Figure 6: Ultrasonic experimental response with electric motors off, for different target materials

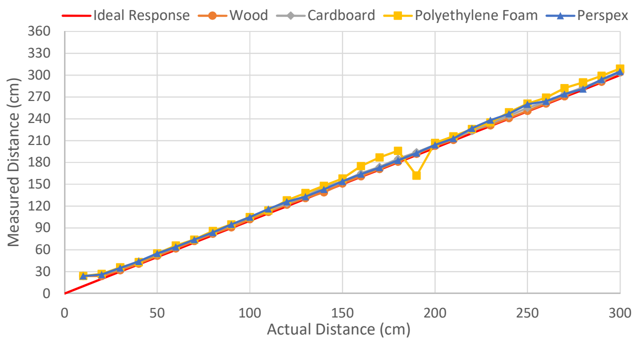


Figure 7: Ultrasonic experimental response with electric motors on, for different target materials

4.3 Infrared Results

Figures 8 and 9 show the distances measured by the infrared sensor in function of the actual distance, for different target materials: wood, cardboard, polyethylene foam and Perspex. The measurements presented in Figure 8 were performed with the MAV motors turned off, while the results presented in Figure 9 were acquired with the MAV motors turned on.

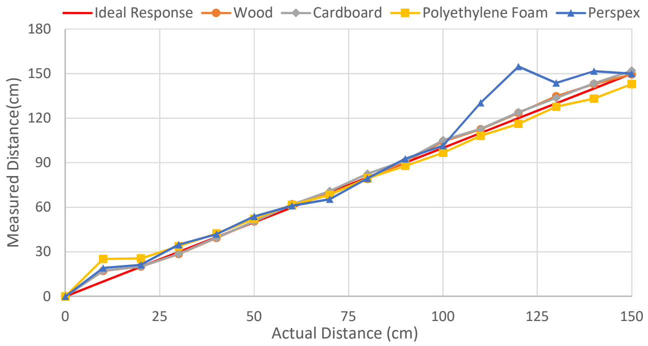


Figure 8: Infrared experimental response with electric motors off, for different target materials

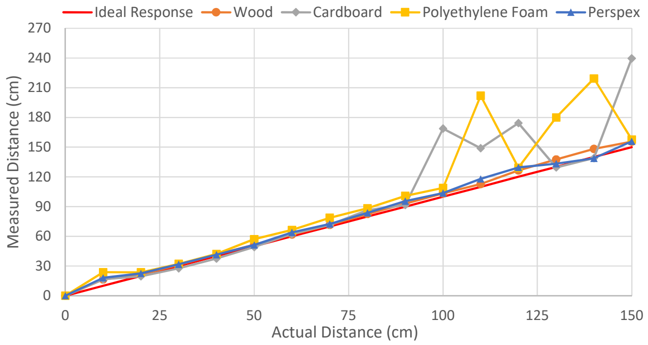


Figure 9: Infrared experimental response with electric motors on, for different target materials

4.4 Statistical Analysis

Considering the measurement results highlighted in the previous subsections, a statistical analysis was performed. Figure 11 shows the RMSE achieved for the sensors (LiDAR, Ultrasonic and Infrared), in relation to each target material. On the other hand, Figure 11 presents the standard deviation of the measurements performed by the sensors.

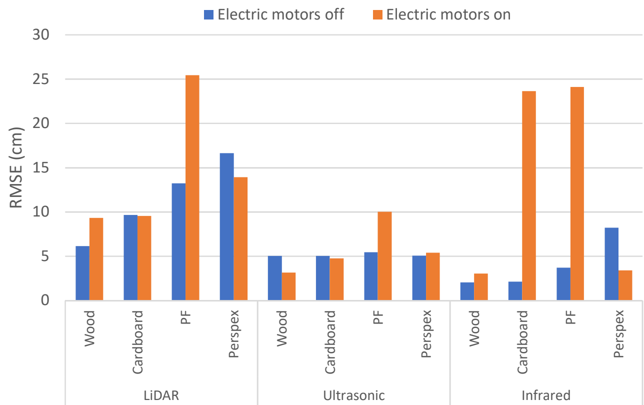


Figure 10: Root mean square error (RMSE) of the measurements.

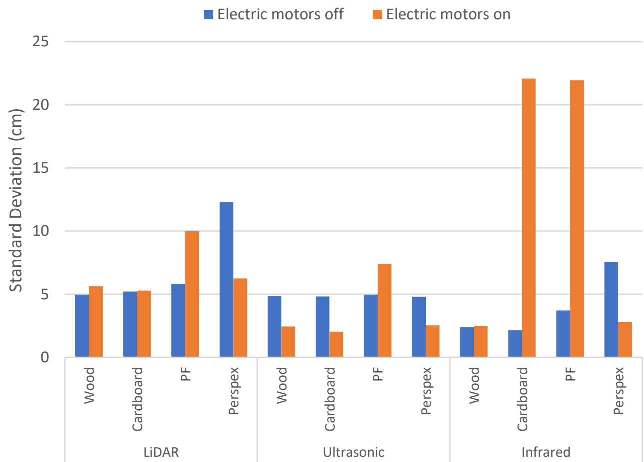


Figure 11: Standard deviation of the measurements.

5 Discussion

Based on the results showed in Section 4, it is possible to extract some conclusions regarding the sensor's performance.

The LiDAR sensor was the one with the worst performance among the three sensors. It provided poorly measurements with high RMSE for all tested scenarios.

In addition to the poor overall performance, when using Perspex as target material it provided the worst measurement (RMSE = 17 cm), among all measurements performed with electric motors powered off. This error was more than two times higher than the error achieved for other target material. Perspex is a highly reflexive material, and the LiDAR working principle takes into consideration the time of flight that a laser pulse takes between its emission and reflection back to the sensor, therefore, having a target with different reflexivity characteristics affects directly the measurement.

On the other hand, the ultrasonic sensor provided the best overall results among all sensors. It provided reliable measurements, with high degree of correlation independently of the distance, target material and presence of disturbances

During the first round of tests, with motors powered off, the ultrasonic sensor provided equivalent performance for all target materials, with similar RMSE and standard deviation. On the other hand, with electric motors powered on, the measurements performed by the ultrasonic sensor remain satisfactorily good for all materials. However, it is noticed a considerably increase in RMSE and standard deviation values obtained for polyethylene foam target. Nonetheless, the archived measurement precision remains acceptable, even in this case. The foam absorbability of sound waves is bigger than the other materials, and since the ultrasonic sensor relies on the time of flight of ultrasonic sound waves, it is expected that it underperforms when it comes to this kind of material.

The infrared sensor showed an excellent response for tests performed with motors powered off. However, the results were considerably affected by the target materials. Besides, the precision of the measurements varies with the measurement distance. For smaller distances, the sensor was able to provide accurate measurements within the tests with and without disturbances, but with bigger distances it provided some unreliable measurements. This unreliable measurements were maximized in the tests with the motors powered on.

6 Conclusion

Nowadays, several solutions are commercially available for distance measurements in embedded applications. When it comes down to MAV design, cost, weight, precision and power consumption of each one of these sensing technologies should considered. Each sensor provides better results according to given boundary conditions, such as: distance range, target material and immunity to disturbances induced by propellers rotation. The experiments performed in this work showed that these 3 factors have great influence on the overall performance of each sensor and, consequently, should be taken in consideration when it comes to selecting the proper technology.

By comparing the RMSE and standard deviation associated to each sensor, it is noticed that the ultrasonic (US) sensor presented the best overall result. Its error remains considerably low for all target materials (always below 10 cm) and it shows good immunity to disturbances produced by propellers rotation. However, it should be mentioned that infrared (IR) sensor outperformed ultrasound in some particular cases, but its behavior is significantly compromised when MAV motors are on. Tables 4 and 5 summarizes the best and worst sensor's performance for each target material in both the conditions tested, with and without disturbances.

Condition	Material			
	Wood	Cardboard	PF	Perspex
Motors off	IR	IR	IR	US
Motors on	IR/US	US	US	IR

Table 4: Sensor with best performance in the tested conditions for different materials

Condition	Material			
	Wood	Cardboard	PF	Perspex
Motors off	LiDAR	LiDAR	LiDAR	LiDAR
Motors on	LiDAR	IR	LiDAR	LiDAR

Table 5: Sensor with worst performance in the tested conditions for different materials

Besides, compared to LiDAR and Infrared technologies, ultrasonic sensors are low power and lightweight devices. The highlighted aspects indicate that, among the analyzed distance sensor technologies, ultrasonic sensors are the most suited devices to be embedded in MAVs for distance measurements.

Acknowledgements

We would like to thank the Brazilian funding agencies CNPq, FAPERJ and CAPES for continued support and supplied resources.

References

[1] Sophia Maharana. Commercial drones. *IJASCET*, 5:96 – 101, 2017.

[2] N. Gageik, P. Benz, and S. Montenegro. Obstacle detection and collision avoidance for a uav with complementary low-cost sensors. *IEEE Access*, 3:599–609, 2015.

[3] V A Zhmud, N O Kondratiev, K A Kuznetsov, V G Trubin, and L V Dimitrov. Application of ultrasonic sensor for measuring distances in robotics. *Journal of Physics: Conference Series*, 1015:32189, 2018.

[4] Jishnu Sankar, S. Adarsh, and K.I. Ramachandran. Performance evaluation of ultrasonic and infrared waves on human body and metal surfaces for mobile robot navigation. *Materials Today: Proceedings*, 5(8, Part 3):16516 – 16525, 2018. International Conference on Advanced Materials (SCICON 16), December 19- 21, 2016.

[5] S Adarsh, S Mohamed Kaleemuddin, Dinesh Bose, and K I Ramachandran. Performance comparison of infrared and ultrasonic sensors for obstacles of different materials in vehicle/ robot navigation applications. *IOP Conference Series: Materials Science and Engineering*, 149(1):012141, 2016.

Bio-inspired Regenerative Flight Trajectory Optimisation Over Flat Topography

N. K. Long,* and S. Watkins†
RMIT University, Plenty Rd, Bundoora VIC 3083, Australia
J.M. Moschetta‡, and V. Bonnin§
ISAE-SUPAERO, 10 Avenue Edouard Belin, Toulouse, 31400, France

ABSTRACT

A gliding technique, known as dynamic soaring (DS), replicates the flight pattern of albatross bird to enable energy-neutral, repeatable flight trajectories. This study investigated the potential for the flight manoeuvers of the albatross to act as a basis for UAV battery power regeneration by means of a windmilling propeller mounted on the aircraft. In order to give an indication of the type of atmospheric and environmental conditions necessary to perform regenerative dynamic soaring (RDS), trajectories were optimized for a small UAV. The optimal flight paths for varying amounts of energy regeneration and periodicity are presented and compared to a base, energy-neutral DS case. The findings suggest that by slightly altering the DS flight pattern, RDS is possible for both open- and closed-loop trajectories with significant battery recharge levels being reached for the UAV modelled under certain conditions.

1 INTRODUCTION

While unmanned flight vehicle technology continues to rapidly expand, new methods for extending their range and endurance are sought out. One solution was found by observing the way that wandering albatrosses (Diomedea Exulans) soar over the ocean seemingly effortlessly over distances exceeding 900km per day [1]. The technique the albatrosses use is known as dynamic soaring (DS). DS uses the vertical wind gradient, such as develops over the ocean in the case of the albatross, to enable energy neutral flight cycles comprised of four flight phases; a climb into the wind (1), a turn from windward to leeward flight direction (2), a descent with the wind (3), and a turn into the wind from leeward flight direction to windward (4) [2] (see Figure 1). By modelling the wind gradient as a logarithmic profile, Sachs was able to accurately simulate the flight pattern of the albatross, optimising the trajectory for minimum wind strength required [3]. Researchers then applied DS control algorithms to drone flight. As an example, Diettert et al. simulated a small UAV flying DS trajectories while minimising the wind strength required, optimising for both open-loop and closed-loop flight paths [4]. Further, Langelaan discusses the potential for increasing small UAV range and endurance by extracting significant amounts of energy from the atmosphere via performing DS, in combination with other soaring techniques, by implementing a control algorithm which monitors atmospheric conditions in real-time [5].

Figure 1 gives an example simulation of the wandering albatross performing DS over a flat ocean. The four flight phases are numbered and colour coordinated, and the trajectory's two-dimensional (2D) projections are depicted on each plane. The orange arrow defines the direction of the wind profile.

Separately, a method for an aircraft to extract the power available in the wind and convert it into usable energy has been discussed in literature. Glauert was the first to examine the potential for placing a windmill on an aircraft [6]. MacCready then reintroduced a similar idea applied to a sailplane [7]. More recently, Barnes conducted a detailed analysis on the functioning of a dual-mode windmilling propeller capable of both generating thrust and extracting wind energy using regenerative braking technologies already employed on cars [8, 9]. The extracted energy was designed to then be stored on an onboard battery via a motor-generator. Bonnin et al. were the first researchers to optimise a flight trajectory for regenerative dynamic soaring (RDS), doing so on the leeward side of a hill where strong wind gradients are known to form [10]. Bonnin introduced a regenerative drag force (D_{gen}) to simulate the additional drag imparted on the aircraft when performing RDS, and optimised a closed-loop trajectory for the least wind strength required while fixing an average battery recharge rate (P_{net}).

This study aims to further the academic research on RDS by establishing the conditions required for RDS over a flat topography. Specifically, the effects of varying P_{net} on the wind strength required will be investigated with the hope of developing a greater understanding of how this effects the resulting trajectory and flight variables.

*Email addresses: *nathan.k.long.91@gmail.com, †simon@rmit.edu.au, ‡jean-marc.moschetta@isae.fr, §bonnin.v@gmail.com

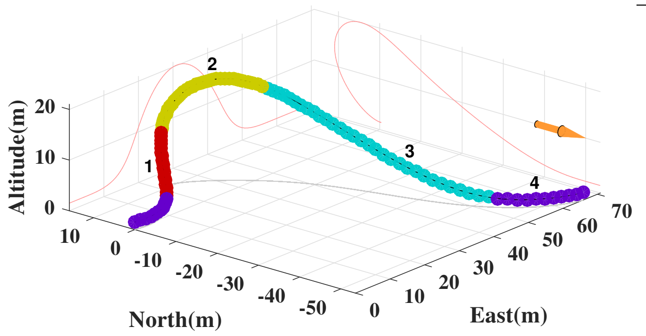


Figure 1: Albatross open-loop dynamic soaring trajectory simulation.

2 Methodology

2.1 Aircraft Model

The aircraft chosen for this study was the DT-18, a small UAV developed by Delair in Toulouse, France. The DT-18 was selected due to its slender body and relatively high aspect ratio (13.1), giving it a construction similar to the general structure of a wandering albatross which is known to have the highest aspect ratio of any living bird (16.8). Furthermore, the DS performance of the DT-18 was thoroughly investigated in Bonnin's doctoral dissertation [11], and used for the RDS optimisation on the leeward side of a hill completed by Bonnin et al. [10].

The aircraft has been represented as a point-mass so as to reduce the complexity of the model and permit a higher degree of abstraction for any conclusions. This point represents the centre of gravity whereby all forces are assumed to act upon it, and bank angle (ϕ) is the only rotation permitted about it.

2.2 Wind Model

Various models have been used to approximate the vertical wind gradient (or wind shear) that develops within the boundary layer over a flat surface, however, Sachs [2], Barnes [9, 8], and Bonnin [11] chose to implement a logarithmic profile for their DS simulations. This decision was based on the flight altitude of the albatross, and on the wind strengths required for DS. One major flaw of this wind profile approximation is its neglect of turbulence-related unsteadiness which would undoubtedly have an effect on the flight dynamics of an aircraft flying within it, however, for the purpose of this study it has been deemed acceptable. Equation 1 gives relationship between the wind strength (V_w) along the logarithmic profile and altitude (z).

$$V_w = U_{ref} \cdot \frac{\ln(z/z_0)}{\ln(z_r/z_0)} \tag{1}$$

Equation 1 also details the reliance on a reference wind height (z_r) and reference wind speed (U_{ref}) in developing the wind profile. A reference wind height of $10m$ was used throughout this study, the same as used by Sachs for his DS study [2]. Furthermore, Equation 1 portrays the effect of surface roughness length (z_0) on the wind profile. A value of $20cm$, representing a flat suburban topography, was fixed for this study in order to focus on the effects of P_{net} on the wind profile and RDS flight variables. The effects of obstacles on the wind profile and trajectories have been assumed negligible. A separate study by Long et al., investigating the effects of surface roughness on RDS flight trajectories, has been **submitted** for publication [12].

2.3 Kinetics, Kinematics, and Thermodynamics

DS has been analysed via both the inertial frame of reference and wind-based frame of reference [13], however, the inertial frame of reference has been selected for the following RDS optimisation as it seems more intuitive to comment on the results based on their ground-relative performance. Based on the inertial reference frame assumption, Sachs et al. argue that the fundamental energy gain experienced during DS is during the turn from upwind to downwind [3].

The coordinate system used for the trajectory mapping is such that the x-axis aligns with north, y-axis with east, and negative z-axis with positive altitude. The relationship between the airspeed vector (V_a) and ground speed vector (V_i), and their associated heading angles (ψ), flight path angles (γ)[1], and ϕ, with respect to the coordinate system are outlined in Bonnin's thesis [11]. Equation 2 then highlights how V_i, V_a and V_w are related.

$$V_a = \sqrt{V_i^2 - 2V_iV_w\cos\psi_i\cos\gamma_i + V_w^2} \tag{2}$$

The system state variables, which define the dynamic state of the aircraft and describe its evolution along a flight trajectory, are V_i, ϕ_i, γ_i, x, y, and z. The control variables, which govern the state variables and are considered manipulable by the aircraft control system or pilot, are the lift coefficient (C_L), ϕ, and the regenerative drag force (D_{gen}) acting on the windmilling propeller.

Figure 2 portrays how D_{gen} is added to the forces acting upon the DT-18 (actual representation of a DT-18 side-view).

Equations 3 to 6 define the force equations for the aircraft system, where the additional drag force (D_{gen}) is of particular interest. It should be noted that D_{gen} then has a subsequent effect on the equations of motion and, thus, the evolution of the aircraft's trajectory. The formulation of the equations of motion is identical to those used by Bonnin [11].

$$\vec{F}_{ext} = F_x\vec{x} + F_y\vec{y} + F_z\vec{z} \tag{3}$$

$$\begin{aligned} F_x = &-L(\sin\phi\sin\psi_a + \cos\phi\cos\psi_a\sin\gamma_a) \\ &-(D + D_{gen})(\cos\gamma_a\cos\psi_a) \end{aligned} \tag{4}$$

$$\begin{aligned} F_y = &\,L(\sin\phi\cos\psi_a - \cos\phi\sin\psi_a\sin\gamma_a) \\ &-(D + D_{gen})(\cos\gamma_a\sin\psi_a) \end{aligned} \tag{5}$$

[1]Subscript a is used to represent air-based variables, and subscript i is used to represent ground-based / inertial variables.

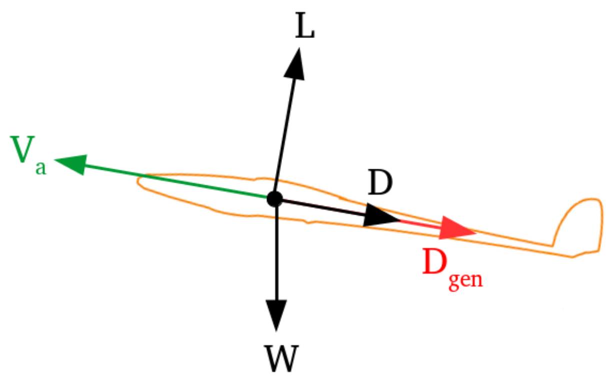


Figure 2: Forces acting on DT-18 UAV developed by Delair (reproduced from Long et al.[12]).

$$F_z = -L\cos\phi\cos\gamma_a + (D + D_{gen})(\sin\gamma_a) + mg \quad (6)$$

RDS requires a series of power conversions in order for the aerodynamic power (P_{aero}) imparted onto the windmilling propeller, defined in Equation 7, to recharge the onboard battery, in Equation 8. This causes efficiency losses to occur and, as such, a fixed efficiency (η_{regen}) of 0.6 is applied to P_{aero} to represent the losses across the entire power chain before entering the battery. This efficiency was based on mimicking the efficiency used in the RDS study by Bonnin et al. [10].

$$P_{aero} = -\vec{D}_{gen} \cdot \vec{V}_a \quad (7)$$

$$P_{gen} = \dot{E}_{bat} = (\vec{D}_{gen} \cdot \vec{V}_a)\,\eta_{regen} \quad (8)$$

$$P_{net} = \frac{\Delta E_{bat}}{t_{cycle}} \quad (9)$$

The net power income (P_{net}), given in Equation 9, represents the average battery recharge rate experienced during one RDS flight cycle. P_{net} is used as a benchmark for the amount of power available for harvesting along a defined RDS trajectory.

2.4 *Optimisation Setup*

The trajectories presented in this study have been optimised by using the Sparse Nonlinear OPTimizer (SNOPT) solver available on the NEOS server. SNOPT takes first-order derivatives as input defined by a specific structure. A Mathematical Programming Language (AMPL) was, thus, used to convert the equations of motion, variable constraints, and objective function into the format required by SNOPT. A MATLAB script coupled to the SNOPT-AMPL package was used as the interface for the data input and output. SNOPT converges on a local optimum solution and, therefore, a large range of input parameters was required to validate the results. The entire setup was replicated from Bonnin's DS trajectory optimisation research [11], and a list of exact constraints for the optimisation can be found within his dissertation. In summary, a series of bounds are placed on the aircraft system which are set to ensure that the resulting trajectories were realistic and physically feasible. Periodicity is also applied as a final condition whereby the variable (z, γ, ψ, ϕ, C_L, V_i, and D_{gen}) final conditions must equate their initial conditions. For the closed-loop optimisations, x and y periodicity was also constrained.

The solver is given an objective function which defines a variable or parameter to either maximise or minimise by manipulating the state and control variables while complying with the constraints and adhering to the equations of motion. As DS is highly dependent on the wind gradient present, the objective function was set to minimising the reference wind speed at a reference height of $10m$ for this study. Identifying the wind strengths required for RDS while varying P_{net} values will help to expand knowledge on the environmental conditions necessary for RDS to be achieved. Furthermore, minimising U_{ref} at $10m$ matches Sachs' optimisation setup [2], and was validated by Bonnin [11] with his setup by using an albatross as the vehicle model and comparing his results to those of Sachs. Differences in results were found to be minor, with a maximum 2% variation found for the variables and parameters tested (U_{ref}, t_{cycle}, Δx_{max}, Δy_{max}, and Δz_{max}).

3 Results and Discussion

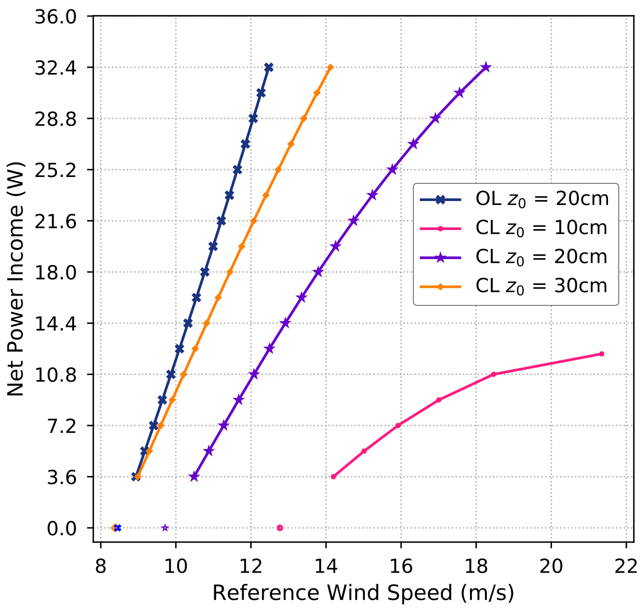


Figure 3: RDS reference wind speed vs. net power income (adapted from Long et al. [12]).

A range of P_{net} values were tested with a fixed surface roughness length of $20cm$ to establish the conditions needed to generate more power during flight.

Figure 3 shows the trends for the U_{ref} required for an enforced P_{net} for both open- and closed-loops. The open-loop trajectory for a surface roughness of $20cm$ is represented by the blue line, requiring the lowest wind strength to permit RDS. The U_{ref} needed was found to increase linearly as P_{net} was increased. The linear relation existed for a closed-loop trajectory over a rougher surface ($z_0 = 30cm$, depicted in orange), however, at a surface roughness of $20cm$ (in purple) the linear trend became distorted at a P_{net} of approximately $18W$, though the aircraft was still able to regenerate the full range of P_{net} values tested. Reducing the surface roughness further to $10cm$ saw a complete breakdown of the linear trend, and an inability of the aircraft to fly closed-loop trajectories beyond a P_{net} value of approximately $13W$. The open-loop RDS trajectory trends were shown to be completely linear down to a surface roughness of $3cm$ [12]. Hereafter, the results from three rates of P_{net}, $10.8W$, $21.6W$, and $32.4W$, are presented alongside the pure DS case for $z_0 = 20cm$.

3.1 Open-Loop Regenerative Dynamic Soaring

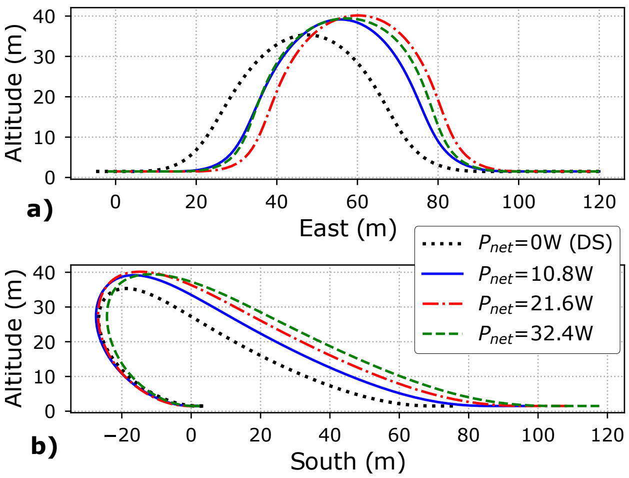


Figure 4: Open-loop 2D trajectories for: a) west-east displacement, and b) north-south displacement.

The west to east displacement (Figure 4a) for the varying P_{net} values, while not following exactly the same paths, were very similar. The DS case, however, significantly varied from the others. The differences between the RDS cases is likely be attributed to SNOPT finding local optimum solutions, not caused by any physical phenomenon. Each trajectory reached an altitude of approximately $40m$ and displaced approximately $120m$ east. The DS case, however, flew to a lower altitude, climbing sooner, and not displacing as far east. This could be explained by the fact that the DS constraints were such that it did not require energy regeneration, only to continue flying, so did not have to climb to reach higher wind strengths.

The comparison of north to south movement shows a further displacement south for higher P_{net} values, and an increase in altitude upon decent.

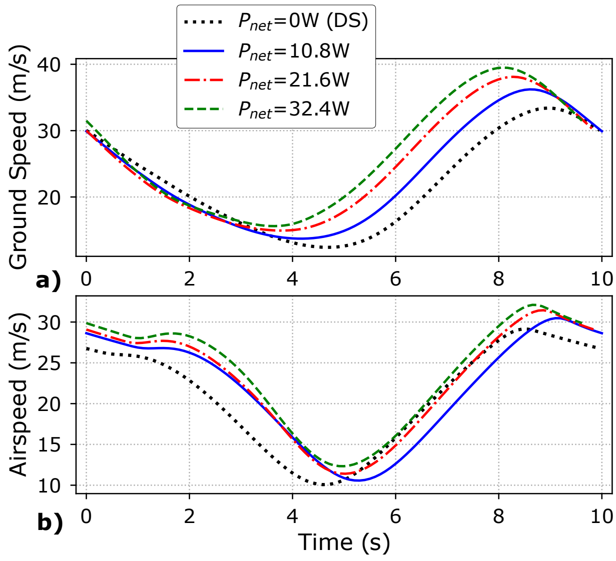


Figure 5: Time versus: a) ground speed. b) airspeed.

Taking the results from Figures 4, and 5a and b together, the further displacement south during descent gave a higher inertial speed, which resulted in a higher airspeed, allowing for the additional power generation (seen in Equation 7).

Figure 6a describes the evolution of D_{gen} during flight. As P_{net} is increased, D_{gen} shifts from being applied only during the descent (as with the $P_{net} = 10.8W$ case) to also being applied during the upper turn from windward direction to leeward direction. As P_{net} is increased the amount of force applied during the upper turn is increased, even surpassing the amount applied during the descent for higher recharge rates (Figure 6a). The maximum airspeeds achievable during decent for the aircraft must be reached whereby no further drag can be accrued, therefore, the aircraft must begin extracting energy during the upper turn. As outlined by Sachs et al. [3], when describing the energy evolution of an aircraft performing DS in the inertial frame, the principal gain in energy is at the top of the trajectory due to the change in orientation of the ground speed vector. Applying extra drag force at the top of the trajectory could be attributed to the fact that the potential energy of the aircraft is high at this point, and with no further power required to ascend or fly windward, the system could afford to convert some of the energy into electricity. The battery recharge rate (P_{gen}) for the $P_{net} = 32.4W$ and $P_{net} = 21.6W$ cases reach their maximum during the descent so, although a higher D_{gen} is applied during the upper turn, a much higher airspeed is seen during descent (Figure 5a).

Figure 6a shows D_{gen} non-dimensionalised with respect to the drag force required for powered straight and level (S&L) flight for maximum endurance by the DT18. D_{gen} ratios are between 60% and 120% of the $S\&L$ aerodynamic

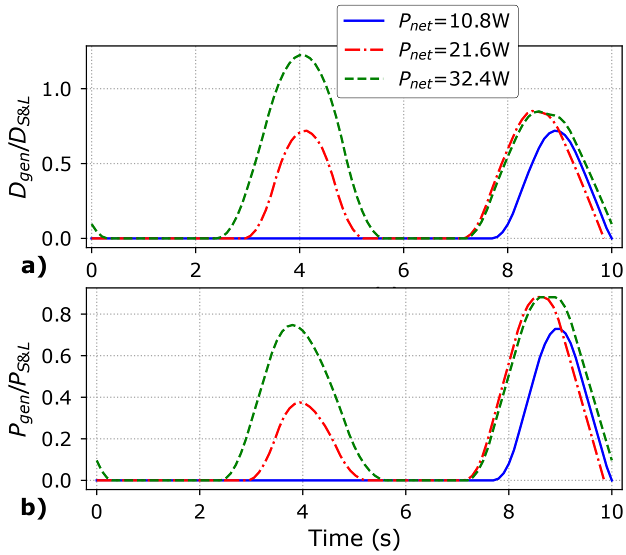


Figure 6: Time versus non-dimensionalised: a) regenerative drag force. b) battery recharge rate.

drag force. Figure 6b gives the variation of P_{gen} with time and has been non-dimensionalised with respect to the power expenditure experienced during maximum endurance $S\&L$ flight. The amount of regenerative power available for harvesting reaches levels close to the same amount of the power required to fly $S\&L$.

3.2 Closed-Loop Regenerative Dynamic Soaring

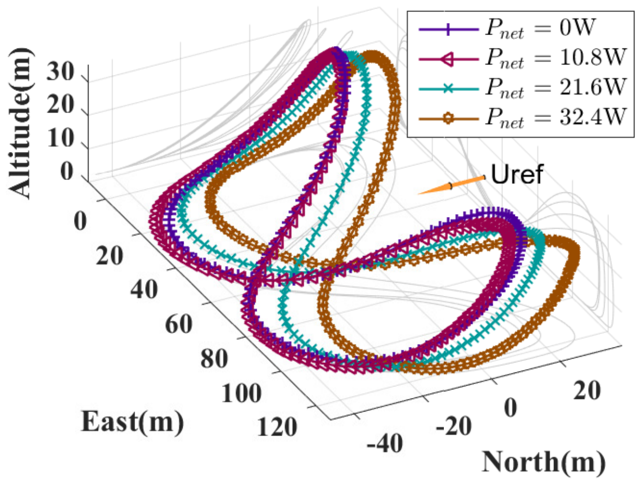


Figure 7: Closed-loop RDS trajectories for varying P_{net} values.

The closed-loop RDS trajectories all followed the same pattern, a recreation of the four DS flight phases which were then mirrored and flown in the reverse direction to form a continual loop. Analyzing the 2D projections in Figure 7 shows that there was a decrease in altitude, a shift upwind, and a increase in the cross-wind displacement for the trajectories with a higher P_{net}. The orange arrow represents the reference wind height and the direction of the wind vector.

As opposed to the open-loop RDS power regeneration, the closed-loop cases harvested no power during the leeward descent; all of the power is regenerated during the upper turns from windward to leeward direction. This could be due to the fact that the aircraft needed to conserve its kinetic energy from the descent to turn back into the wind and climb again without any net displacement with the wind.

4 Conclusion

The investigation into the effects of increasing P_{net} on RDS flight trajectories showed that, for a higher P_{net}, a greater area for displacement was required. Furthermore, for the open-loop RDS trajectories, power harvesting shifted from only during flight descent for lower P_{net} values, to also occurring during the upper turn for higher P_{net} values. On the other hand, the closed-loop trajectories extracted power from the wind only during the upper turns from windward flight to leeward flight, and the trajectories experienced a general flattening for higher levels of P_{net}.

Future studies into RDS should focus on the inclusion of thrust augmentation to test how a combined windmilling propeller would affect the RDS flight trajectories. The model should also be elaborated to investigate how the aerodynamic performance of different windmilling propellers will affect RDS performance, and further refinement of the wind profiles should be incorporated into RDS optimisations to represent the effects of obstacles and local unsteadiness.

Acknowledgements

The authors would like to express their appreciation for the Department of Aerodynamics, Energetics and Propulsion (DAEP) at ISAE-SUPAERO for their support and funding of this research, and to RMIT University for their input and backing of the project. A special thanks would like to be given to Dr Thierry Jardin and Yuchen Leng at the DAEP for their regular advice and assistance throughout the study, as well as to Dr Abdulghani Mohamed at RMIT for his encouragement and guidance.

References

[1] S. M. Waugh, P. A. Prince, and H. Weimerskirch. Geographical variation in morphometry of black-browed and grey-headed albatrosses from four sites. *Polar Biology*, 22(3):189–194, 1999.

[2] G. Sachs. Minimum shear wind strength required for dynamic soaring of albatrosses. *Ibis*, 147(1):1–10, 2005.

[3] G. Sachs, J. Traugott, A. P. Nesterova, F. Kummeth, W. Heidrich, A. L. Vyssotski, and F. Bonadonna. Fly-

ing at no mechanical energy cost; disclosing the secret of wandering albatross. *PLoS One*, 7(9), 2012.

[4] M. Deittert, A. Richards, C. A. Toomer, and A. Pipe. Engineless unmanned aerial vehicle propulsion by dynamic soaring. *Journal of Guidance, Control, and Dynamics*, 32(5):1446–1457, 2009.

[5] J. W. Langelaan. Power generation and energy management. *Encyclopedia of Aerospace Engineering*, pages 1–20, 2016.

[6] H. Glauert. *The elements of aerofoils and airscrew theory*. Cambridge University Press, Cambridge, United Kingdom, 1926.

[7] P. MacCready. Regeneratve battery-augmented soaring. *Technical Soaring*, 23(1):28–32, 1999.

[8] P. J. Barnes. Aircraft energy gain from an atmosphere in motion: dynamic soaring and regen-electric flight compared. 2015.

[9] P. J. Barnes. Regenerative electric flight: Synergy and integration of dual-role machines. In *53rd AIAA Aerospace Sciences Meeting*, 2015.

[10] V. Bonnin, E. Benard, and J-M. Moschetta. Energy-harvesting by dynamic soaring on the leeward side of a hill. 2016.

[11] V. Bonnin. *From albatrosses to long range UAV flight by dynamic soaring*. PhD thesis, 2015.

[12] N. K. Long, S. Watkins, J.M. Moschetta, and V. Bonnin. Regenerative dynamic soaring trajectory augmentation over flat terrains. to appear.

[13] D-N. Liu, Z-X. Hou, Z. Guo, X-X. Yang, and X-Z. Gao. Bio-inspired energy-harvesting mechanisms and patterns of dynamic soaring. *Bioinspiration Biomimetics*, 12(1):016014, 2017.

Bibliography

[1] Joost Meulenbeld, Christophe De Wagter, and Bart Remes. Modeling delftacopter from flight test data. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 18–29, Melbourne, Australia, Nov 2018. 13

[2] Christophe De Wagter, Bart Remes, Rick Ruijsink, Erik van der Horst, Freek van Tienen, Denis van Wijngaarden, Joost Meulenbeld, and Kevin van Hecke. Delftacopter propulsion optimization from hover to fast forward flight using windtunnel measurements. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 30–38, Melbourne, Australia, Nov 2018. 13

[3] Leandro Diniz de Jesus, Alexandre Ramos Brandão, Caique de S. F. Duarte, and Felix Mora-Camino. Increased autonomy for uav using solar panels and take advantage of updrafts through algorithm. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 39–44, Melbourne, Australia, Nov 2018. 13

[4] Albert Medina and Matthew Rockwood. On the response of leading-edge phenomena and near-wake formations to trailing-edge flap actuation. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 45–53, Melbourne, Australia, Nov 2018. 13

[5] Shupeng Lai, Menglu Lan, and Ben M. Chen. Safe corridor based task interface for quadrotors. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 54–59, Melbourne, Australia, Nov 2018. 13

[6] Martin Lefebvre, Renaud Kiefer, Marc Vedrines, and Hicham Chibane. Carbon fibre/pvc foam sandwich composite modelization for mavs & long range drones structures. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 60–65, Melbourne, Australia, Nov 2018. 13

[7] Thiago Rodrigues da Costa, Marcos Pereira Caldas, Pablo Milheiro Novaes de Araújo, and Eduardo Costa da Silva. Topological optimization applied towards thedevelopment of a small and lightweight mav composite frame. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 66–71, Melbourne, Australia, Nov 2018. 13

[8] Inrae Kim, Heetae Park, Seungkeun Kim, Jinyoung Suk, and Sungbong Cho. Disturbance observer based control design of flapping-wing mav considering model uncertainties. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 72–80, Melbourne, Australia, Nov 2018. 13

[9] Seong Lok Nam, Steven Aurecianus, Hoang Vu Phan, Taesam Kang, and Hoon Cheol Park. Experimental study on the phase delay of low-cost imu, low pass, and kalman filter and its effect on the phase margin of angle estimation. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 81–85, Melbourne, Australia, Nov 2018. 13

[10] Fabian Binz, Tobias Islam, and Dieter Moormann. Attitude control of tiltwing aircraft using a wing-fixed coordinate system and incremental nonlinear dynamic inversion. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 86–93, Melbourne, Australia, Nov 2018. 13

[11] Aaron Lopez, José Martinez, and Israel Cruz. Towards micro aerial manipulation using a computational compensation strategy. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 94–99, Melbourne, Australia, Nov 2018. 13

[12] Torbjørn Cunis and Elgiz Baskaya. Performance of unmanned aircrafts in the event of loss-of-control. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 100–105, Melbourne, Australia, Nov 2018. 13

[13] Jeremy Cox and K.C. Wong. Predictive feedback augmentation for manual control of a uav with latency. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 106–113, Melbourne, Australia, Nov 2018. 13

[14] Seongyoung Kim, Myungkang Kim, Seungkeun Kim, and Jinyoung Suk. Design, fabrication, and flight test of articulated ornithopter. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 114–119, Melbourne, Australia, Nov 2018. 13

[15] Diana Olejnik, Aadithya Sujit, Matej Karasek, Bart Remes, and Guido de Croon. Wing sweeping mechanism for active control and stabilisation of a flapping wing mav. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 120–126, Melbourne, Australia, Nov 2018. 13

[16] Jong-Wan Lee, Jong-Seob Han, and Jae-Hung Han. Forward flight control analysis of bioinspired flapping-wing micro air vehicle. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 127–132, Melbourne, Australia, Nov 2018. 13

[17] Zhen Liu, Penglei Che, Tianjiao Dang, Jun Li, and Chen Bu. Propulsion performance investigation of bio-inspired nano rotor base on fluid-structure interaction. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 133–143, Melbourne, Australia, Nov 2018. 13

[18] Longjiang Li, Hongmei Zhou, Jianjun Yang, and Yuming Mao. Force-directed formation and guidance for large scale swarming system of micro air vehicles. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 144–149, Melbourne, Australia, Nov 2018. 13

[19] Aurelien Cabarbaye, Titouan Verdu, Jacson Miguel Olszanecki Barth, and Gautier Hattenberger. Accurate control law for low-cost uav. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 150–159, Melbourne, Australia, Nov 2018. 13

[20] Sebastien Prothin, Claudia Fernandez Escudero, Thierry Jardin, and Nicolas Doué. Archean: Aerodynamics of rotors in confined environments - study in ground and corner effect. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 160–173, Melbourne, Australia, Nov 2018. 13

[21] Nay Lin Oo, Nicholas Kay, Angus Brenkley, and Rajnish Sharma. Investigation into the behaviour of an iced low reynolds number aerofoil. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 174–179, Melbourne, Australia, Nov 2018. 13

[22] Pengfei Yu, Derrick Ho, and K.C. Wong. An adaptable indoor flight test implementation for small uavs. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 180–185, Melbourne, Australia, Nov 2018. 13

[23] Taegyun Kim, Hyemin Mo, Seungkeun Kim, and Jinyoung Suk. Simulation & flight test of ducted-fan uavs for formation flight. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 186–191, Melbourne, Australia, Nov 2018. 13

[24] Hoang Vu Phan, Steven Aurecianus, Taesam Kang, and Hoon Cheol Park. Attitude control mechanism in an insect-like tailless two-winged flying robot by simultaneous modulation of stroke plane and wing twist. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 192–197, Melbourne, Australia, Nov 2018. 13

[25] Kim Loan Au and Hoon Cheol Park. Aerodynamics during forward flight of a tailless flapping-wing micro air vehicle. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 198–203, Melbourne, Australia, Nov 2018. 13

[26] Thierry Jardin, Sebastien Prothin, Philippe Barricau, and Jean-Renaud Frutos. Development of a 200 mg bio-inspired nano-flying robot. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 204–210, Melbourne, Australia, Nov 2018. 13

[27] Gakuki Ise, Daiki Yoda, Chinami Takiguchi, Naoya Shigematsu, Tomohiro Shiratori, Hisami Nakano, Woojae Jeon, Taichi Nozawa, Kei Iwashiro, Takumi Nishino, Toshiatsu Munakata, and Takanobu Watanabe. Controlling the aerial posture of a flapping-wing micro air vehicle by shifting its centre of gravity. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 211–215, Melbourne, Australia, Nov 2018. 13

[28] Steven Aurecianus, Hoang Vu Phan, Seong Lok Nam, Taesam Kang, and Hoon Cheol Park. Flapping wing micro air vehicle (fw-mav) state estimation and control with heading and altitude hold. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 216–221, Melbourne, Australia, Nov 2018. 13

[29] Martin Heinold and Christian J. Kähler. Aerodynamic investigation of the free flapping flight of a saker falcon using a 3d multi-view reconstruction method. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 222–236, Melbourne, Australia, Nov 2018. 13

[30] Gautier Hattenberger. Precision landing for fixed-wing uav using ultra-wide-band ranging. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 237–242, Melbourne, Australia, Nov 2018. 13

[31] L. Oyuki Rojas-Perez, Roberto Munguia-Silva, and José Martinez-Carranza. Real-time landing zone detection for uavs using single aerial images. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 243–248, Melbourne, Australia, Nov 2018. 13

[32] Graham Dorrington. On the scaling of dragonflies. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 249–255, Melbourne, Australia, Nov 2018. 13

[33] Zihao Wang and K.C. Wong. Autonomous bird deterrent system for vineyards using multiple bio-inspired unmanned aerial vehicle. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 256–261, Melbourne, Australia, Nov 2018. 13

[34] Lucas de Rijk, Seumas Beedie, and Karl Stol. Precision weed spraying using a multirotor uav. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 262–267, Melbourne, Australia, Nov 2018. 13

[35] Luciano Ribeiro, Caique de S. F. Duarte, Alexandre Ramos, and Felix Mora-Camino. Visual servo control in quadrotor for power line tracking. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 268–272, Melbourne, Australia, Nov 2018. 13

[36] Christina Andrikopoulou, Stéphane Le Calvé, Christophe Sutter, Florian Noël, Alberto Rodríguez-Cuevas, Irene Lara-Ibeas, Renaud Kiefer, Marc Vedrines, Thomas Pavot, Martin Lefebvre, Jörg Ettrich, Werner Schröder, Stefan Junk, and Steffen Schrock. Mav payload: An air-quality monitoring system for integration inside a drone. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 273–276, Melbourne, Australia, Nov 2018. 13

[37] Luis González-Guzmán and José Martinez-Carranza. Automatic combination of line and point descriptors for thermal aerial image mosaicing. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 277–282, Melbourne, Australia, Nov 2018. 13

[38] René Parlange, José Martinez-Carranza, Luis Enrique Sucar, and Beibei Ren. Vision-based autonomous navigation for wind turbine inspection using an unmanned aerial vehicle. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 283–288, Melbourne, Australia, Nov 2018. 13

[39] Titus Braber, Christophe De Wagter, Guido de Croon, and Robert Babuska. Optical-flow-based stabilization of micro air vehicles without scaling sensors. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 289–297, Melbourne, Australia, Nov 2018. 13

[40] Pengfei Wang, Zhi Gao, Hailong Qin, Myo Tun Aung, Xudong Cheng, Mingjie Lao, Feng Lin, and Swee Huat Rodney Teo. Real-time disparity map reconstruction with on-board fpga by semi-global matching and weighted least square filtering. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 298–302, Melbourne, Australia, Nov 2018. 13

[41] Bumsoo Park and Hyondong Oh. Obstacle avoidance for uavs via imitation learning from human data with 3d-convolutional neural networks. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 303–308, Melbourne, Australia, Nov 2018. 13

[42] Manoel Feliciano da Silva Neto, Guilherme Siqueira Eduardo, Eduardo Costa da Silva, and Wouter Caarls. Computer vision based solutions for mav target detection and flight control. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 309–314, Melbourne, Australia, Nov 2018. 13

[43] B.P. Duisterhof and Guido de Croon. Autonomous landing algorithm using a sun position predicting model for extended use of solar powered uavs. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 315–323, Melbourne, Australia, Nov 2018. 13

[44] Kelvin Amancio, Gabriel Pìnheiro, Lucas Teixeira, and Caique de S. F. Duarte. Enabling intelligent and autonomous drones using embedded linux. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 324–327, Melbourne, Australia, Nov 2018. 13

[45] Sean Morrison, Alex Fisher, and Fabio Zambetta. Towards intelligent aircraft through deep reinforcement learning. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 328–335, Melbourne, Australia, Nov 2018. 13

[46] Hoijo Jeaong, Seunghwan Jo, Seungkeun Kim, Jinyoung Suk, and Yung-Gyo Lee. Simulation and flight experiment of a quadrotor using disturbance observer based control. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 336–347, Melbourne, Australia, Nov 2018. 13

[47] Renan de Lima Simões Mondego Vilela, Henrique Pinheiro Saraiva, Erico de Souza Prado Lopes, Rodrigo Camargo Martins Pires, and Eduardo Costa da Silva. Influence of the target material and disturbance sources in the accuracy of distance sensors for mav applications. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 348–353, Melbourne, Australia, Nov 2018. 13

[48] Nathan Long, Simon Watkins, Jean-Marc Moschetta, and Vincent Bonnin. Bio-inspired regenerative flight trajectory optimisation over flat topography. In S. Watkins, editor, *10th International Micro Air Vehicle Competition and Conference*, pages 354–359, Melbourne, Australia, Nov 2018. 13