IMAV2013



International Micro Air Vehicle Conference and Flight Competition

Toulouse, France, September 17-20, 2013

http://www.imavs.org/

Proceedings

of the

International Micro Air Vehicle Conference and Flight Competition

IMAV 2013

September 17 – 20, 2013
Toulouse, France



http://www.imavs.org/

Scientific Session 1

Aerodynamics and propulsion performances

Chairman : Farid Zizi

- Murat Bronz, Gautier Hattenberger and Jean-Marc Moschetta. ***Development of a Long Endurance Mini-UAV : ETERNITY***
- Klaus-Peter Prof. Dr.-Ing. Neitzke. ***Rotary Wing Micro Air Vehicle Endurance***
- Peng Lv, Sebastien Prothin, Fazila Mohd-Zawawi, Emmanuel Benard, Joseph Morlier and Jean-Marc Moschetta. ***Adaptive proprotors as applied to convertible MAVs***

http://www.imavs.org/

Development of a Long Endurance Mini-UAV : ETERNITY

Murat Bronz[1]*, Gautier Hattenberger[1]† and Jean-Marc Moschetta[2]‡

[1] l'Ecole National de l'Aviation Civil, Toulouse, France
murat.bronz@enac.fr, gautier.hattenberger@enac.fr
[2] Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse, France
jean-marc.moschetta@isae.fr

Abstract

This study presents the effort given for the first prototype of a *Long Endurance Mini UAV* concept called *Eternity*. A multi-disciplinary conceptual aircraft design program called *CDSGN* is developed and used for the design of the *Eternity*. Unlike the traditional design methods that uses statistical data from the previous well-flown aircrafts, *CDSGN* analyses numerous aircraft candidates and simulates each candidate for the given mission definition and outputs the corresponding performance. The unique property of the presented design methodology comes from a computationally fast and physically accurate modelling of the aerodynamic characteristics of each candidate by using a modified version of a vortex lattice program called AVL from Mark Drela. Two types of configurations have been analysed for the *Eternity* design, conventional and flying-wing. A wide envelope of variable design parameters used for both configurations such as wing surface area, cruise speed, battery capacity, different airfoils, etc... Integration of solar cells, and the management of solar energy is also considered for every candidate. Only the wing span size has been limited to one meter. Additionally, the on-board avionics and payload weights and sizes are fixed for every candidate as they are independent of the design. Analyses by *CDSGN* concluded the dominance of the conventional configuration for the given long endurance mission performance both on solar and non-solar conditions. Optimum wing surface area and the on-board battery energy found interactively by a post-filtering program developed in-house. A custom airfoil family, transitioning along the span, have been designed specifically for the corresponding local Reynolds number for specific spanwise locations. A wind-tunnel campaign is performed with a full-scale model and first flight tests have been performed in order to show the feasibility of long endurance flights.

http://www.imavs.org/

1 Introduction

Endurance performance enhancement of mini and micro UAVs has been a common interest. The endurance performance relies on many aspect of the design where it has to be concentrated in a more specific way than the traditional conceptual design methods such as Raymer[9] and Roskam[10].

This study will present the design optimization of a one-meter UAV, called *Eternity*, which shows the feasibility of an electric powered Long-Endurance MAV concept. The particular interest is on the developed CDSGN conceptual aircraft design tool specialized on MAV scale. Each of the subsystems (e.g. aerodynamics,structure, propulsion,control,...) modelled seperately in order to see their final effect on the performance of the vehicle.

*Assistant Professor in Applied Aerodynamics
†Assistant Professor in Flight Dynamics
‡Professor in Aerodynamics

2 CDSGN : The conceptual design tool

CDSGN [4] is a multi-disciplinary conceptual aircraft design program. The core aerodynamic calculations are done by using a modified version of AVL program [7], which uses vortex lattice method. Modifications include the airfoil viscous drag addition coming from an XFOIL[6] database, fuselage skin friction drag with flat plate approximation [8], and stall information according to the maximum local lift coefficients of the particular airfoils at corresponding Reynolds number.

It uses a computationally fast, yet physically realistic method whose results have been verified with several wind-tunnel and flight tests[4, 3]. Figure 1 shows the brief flow chart of the program.

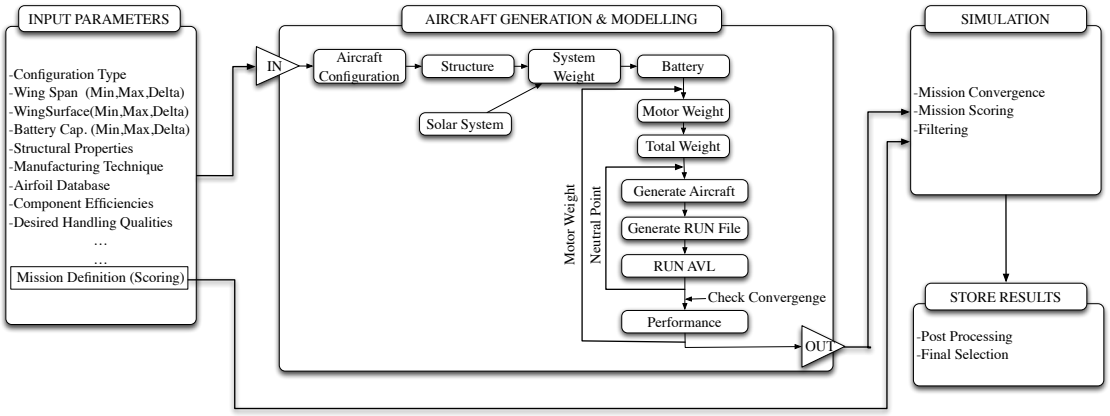


Figure 1: Brief flowchart of *CDSGN* program.

3 Design Envelope

In order to see the endurance and range performance limits of a one-meter aircraft, various configuration "*cases*" are investigated by CDSGN. Table 1 shows the main variables and their ranges used as an input on the program. Each of these variables is examined on different aircraft configurations with different airfoils, wing planforms, energy systems and stability characteristics. This facilitates the performance comparison of each *case* for the whole variable space. Comparisons will be explained in the related sections.

Main Variables	Range	
Aircraft Configuration	Conventional or Flying-Wing	
Energy Source	Battery only or +Solar energy	
Wing Surface Area	$0.05 - 0.2$	m^2
Flight Speed	$8 - 20$	m/s
Battery Energy	$20 - 210$	Wh

Table 1: The range of main variables used for the design envelope of *Eternity*.

http://www.imavs.org/

3.1 Configuration Selection

Although several aircraft configurations exist, in order to keep the design simple and parametric over the main variables shown in table 1, only conventional and flying wing configurations are considered. Each aircraft is generated automatically by *CDSGN* for each single main variable value. Figure 2 shows some of the automatically generated aircraft configurations as an example.

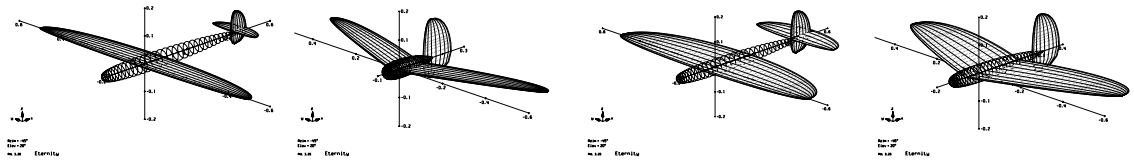


Figure 2: Examples of some automatic generated aircraft configurations by *CDSGN*.

The main objective of this design phase is to see the high-end performance capabilities of each configuration.

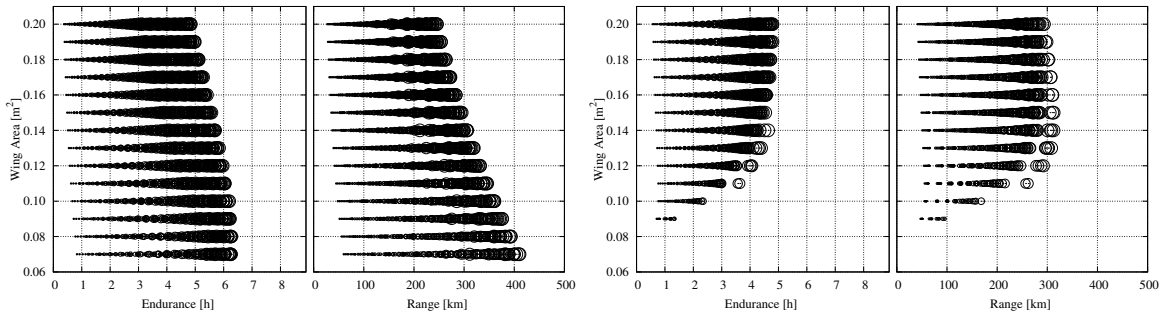


Figure 3: Performance plot of 1-meter conventional aircraft on the left and 1-meter flying wing aircraft on the right in various specifications with only battery. Circle radius represents the relative on-board energy being between $20\,Wh$ and $210\,Wh$ for min and max size.

Figure 3 shows the theoretical performance potentials of 1-meter conventional aircraft (on the left) within the initial specified design constants.The on-board energy, relatively represented by the circle radius, varies between $20\,Wh - 210\,Wh$ corresponding to $0.1\,kg - 1.5\,kg$ pure battery mass respectively. It can be seen that as the wing area increases, the saturation of maximum on-board energy reduces the maximum endurance and range performance. This was an expected behaviour; the performance of greater surface area configurations can easily be increased by increasing the maximum on-board energy limitation. However, the ease of operation (like hand launching) and certain UAV regulations favour keeping the total mass of the UAV under $2\,kg$ [2].

On the right of Figure 3, the same type of performance plot for flying wing configuration is presented. In contrast to the conventional configuration, flying wing configuration tends to have better endurance and range performance for increased wing area. This is mainly due to having a lower maximum lift coefficient C_{Lmax} in trimmed level flight conditions, in comparison to conventional configuration because of airfoil type and not having a separate horizontal tail.

http://www.imavs.org/

Effect of Solar Energy

In order to see the effect of using Solar Energy, the same type of analysis was done while taking into account the additional weight and limitations of solar cells, such as the maximum power point tracker board, connection cables, etc.

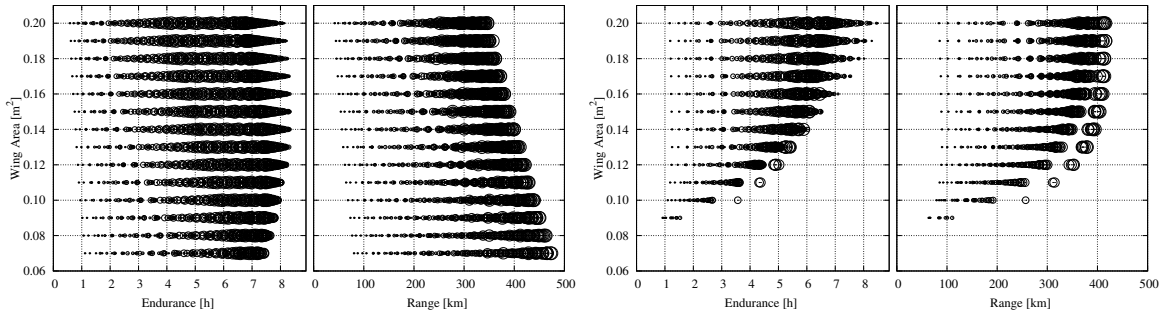


Figure 4: Performance plot of 1-meter conventional aircraft on the left and 1-meter flying wing aircraft on the right in various specifications with battery and solar cells. Circle radius represents the relative on-board energy being between $20\,Wh$ and $210\,Wh$ for min and max size.

Figure 4 shows the endurance and range performance of 1-meter conventional (on the left) and flying-wing (on the right) configuration with on-board battery and solar cells. For the conventional configuration, the increase of both endurance and range performance is clearly visible if it is compared with figure 3. Solar energy enhances the endurance up to $8+\,hours$, where it was around $6\,hours$ before. The range increases $100-150\,km$ on average. Additionally, maximum endurance is evenly distributed over the various wing areas. The need for additional on-board energy is still visible for the bigger wing areas, but this time with the effect of solar cells covering the wing area, it is possible to use a smaller amount of on-board energy in order to be light. This ensures reduction of required flight power, which is maintained by the solar cells on the big wing surface. This is the reason why we see the configurations with smaller on-board energy having better endurance performance for the bigger wing surfaces, although the bigger wing creates more drag than the smaller wings at the same flight speed.

For the flying-wing configuration, the endurance performance seems to increase from $4.5\,hours$ to $8\,hours$ in certain *cases*. The *cases* with bigger wing areas, which carry small amounts of on-board energy, have slower cruise speeds and obtain more advantages from the solar energy. It should still be noted that these *cases* are too sensitive for flight conditions and environment such as a small amount of difference in the cruise flight speed or a decrease in solar irradiation, which end up shifting the performance values dramatically.

Conventional or Flying-Wing ?

Previous figures showed that both conventional and flying wing configurations are able to achieve $8+\,hours$ of flight time with the help of solar energy. However, flying wing configuration is only able to reach that goal for a limited number of *cases*, while conventional configuration has well proven that the goal is reachable for a wide range of design variables. This is mainly due to the airfoil performance and the longitudinal trim issues that are limited with the flying wing configuration, especially at higher lift coefficients. This proves that the conventional configuration will be more robust for a real-

http://www.imavs.org/

world application, where the conditions and environment change constantly. Additionally, non-solar performance of conventional configurations highly outperformed flying wing configurations on average. While making these conclusions, the whole envelope of design variables has taken into account for both configurations.Finally, the conventional configuration is selected as it outperformed the flying-wing configuration in these mission requirements.

3.2 Required Optimum Energy (On-Board)

The weight of the stored on-board energy is one of the key points in enhancing the platform performances, so it is important to select the optimum on-board energy that will give the best range and endurance performances for both solar and non-solar conditions. Table 2 shows the general specifications of off-the-shelf batteries. Lithium-Polymer batteries are used in all of the analyses, since Lithium-Sulfur batteries are almost impossible to obtain for our project because of their supplier policies(Sion Power [1]). As a result of several discharge tests, the specific energy is taken as 190 Wh/kg different from the general table.

		Ni-Cd	Ni-Mh	Li-Po	Li-S
Specific Energy	(Wh/kg)	40	80	180	350
Energy Density	(Wh/l)	100	300	300	350
Specific Power	(W/kg)	300	900	2800	600

Table 2: Typical battery specifications.

In order to see the effect of on-board energy on performance, the same analyses done previously are plotted in a different way. Figure 5 shows the endurance and range performance of different designs for non-solar configuration (on the left) and solar configuration (on the right) versus on-board energy. For non solar configuration, the effect of additional on-board energy is getting to a saturation around 150 Wh , and above that value carrying more energy on board has no advantage for endurance performance. For the range performance, although there is a reduction on the increase, it keeps increasing with the additional energy. Caution should be given to the point that the given battery volume after a certain value will be impossible to carry on-board due to the size of the plane.

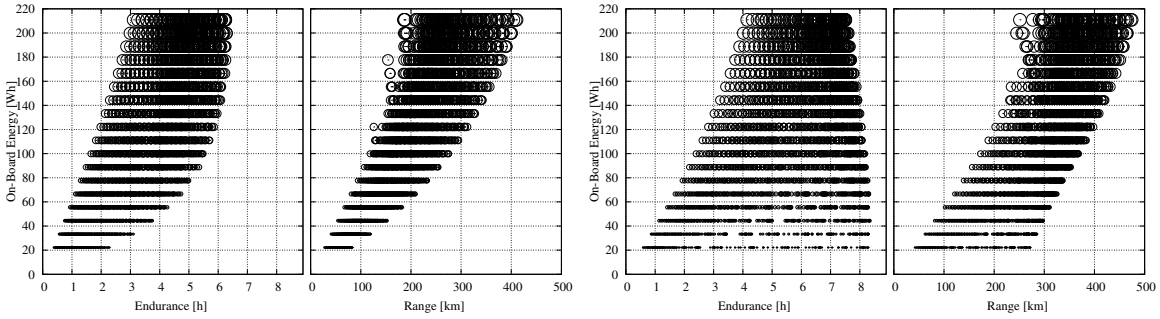


Figure 5: Effect of on-board energy on endurance and range performance, on the left with battery-only and on the right with solar energy addition.

On the right of figure 5, for solar configurations, it is assumed that 70% of the wing area is covered with solar cells. As seen, solar energy has a big influence on endurance performance, especially for the

http://www.imavs.org/

configurations that have low on-board energy, thus lower weight. This result was already presented in the previous analysis, but the clear relation with the on-board energy has just been shown again. For solar configurations, an optimum on-board energy of $50\,Wh$ gives the best endurance performance. In these calculations, the specific energy of the battery was chosen as $190\,Wh/kg$, which results in around $250\,gr$ of battery weight for maximum endurance performance of a $1\,m$ spanned aircraft in solar conditions. For the range performance, the behaviour is not different from the non-solar flight results. The more on-board energy that is carried on the aircraft, the better the range performance.

4 Design Philosophy

It was shown in the previous section that one configuration will not be capable of having the optimum performance in both endurance and range performance at solar and non-solar conditions. Designing the aircraft for only one particular condition will make it fail during almost all flight conditions different from the design point. The main focus of the design is to have a long-endurance aircraft, and the design philosophy is to obtain long-endurance performance in the operational cases as well, and not only on one particular condition.

4.1 Idea of Variable Configuration

It is possible to design different aircraft for different missions and conditions, however *Eternity* long endurance mini UAV should be as compact as possible. Therefore, concentrating on one wing planform which can be used with two different fuselages and various battery packages was found to be the most promising option. Finally, there will be different systems that will be appropriate for various missions and flight conditions.

The battery pack is designed as a wing joiner simultaneously. The sizing of the joiner battery pack is also a key point, as it plays a big role on the wing planform and airfoil limitations. AMI cell [1] ($155mm \times 60mm \times 4mm$), $4.5\,Ah$ $3.7\,V$ seemed to be the best fit for the wing joiner battery pack. Finally, a four cell battery pack is built in the joiner.

5 Design Details

With the battery cell type selected, the on-board energy was partially decided as a result. The wing joiner will consist of a four cell $4.5\,Ah$ ($66\,Wh$) battery, which is suitable for long endurance mission in a sunny day. For overcast weather or a long range mission, it is preferable to use 2 packs ($132\,Wh$) of batteries. In order to fit an additional battery pack into the system, the fuselage has to be enlarged. This is not going to be beneficial for the long endurance version, as it will cause some additional drag. These points indicate that two separate fuselages are needed in order to optimize the system for both endurance and range missions using the same wing and tail parts. This decision also makes it possible to use different propulsion systems (propeller, motor, speed controller) optimized particularly for each mission, rather than compromising between them.

Figures 6 and 7 shows the endurance performance for various wing areas and flight speeds for $66\,Wh$ and $132\,Wh$ on-board energy in both solar and non-solar conditions. The same performance evoluation is done for the range and taking into account the four conditions and additional operational requirements (such as take-off and landing), $0.14\,m^2$ of wing area performs the best.

[1] http://www.amicell.co.il/

http://www.imavs.org/

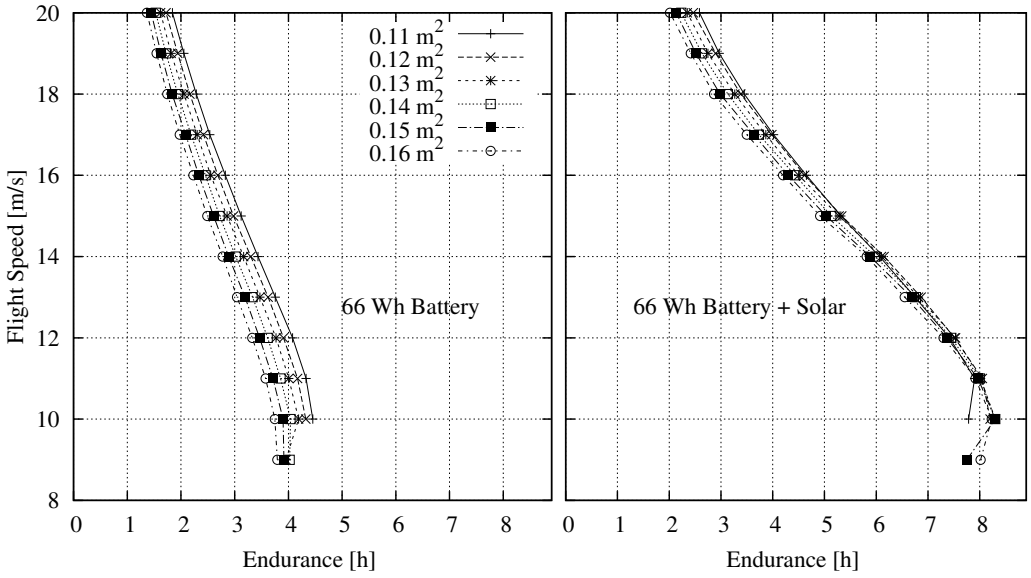


Figure 6: The effect of various flight speed and wing area on the endurance performance of non-solar and solar configurations with 66Wh of on-board energy.

5.1 Design of the Airfoil Family

In the conceptual design phase, suitable airfoils were selected as candidates by examining their maximum power coefficient ($C_l^{1.5}/C_d$) values. Then, the effect of different airfoils on endurance and range performance are investigated. Finally, by using the best existing airfoil as a starting geometry, a custom airfoil is designed by using XFOIL[2] according to the main requirements of the design. The author would like to thank Philip Kolb for contributing to this very important phase with his knowledge and experience. PKMB500 airfoil is designed by Philip Kolb, which outperformed all previous airfoils existing in the selected database for our application.

The main reason why PKMB500 can give better results than its precedents relies on the fact that it is designed according to the *Eternity*'s specifications. The wing area was designated $0.145\,m^2$, and the battery weights were determined by the selected optimum capacities for solar and non-solar conditions. Since almost all of the main parameters of the design have been selected, it is easy to calculate the $Re\sqrt{C_L}$ of the mean aerodynamic chord via equation 1 and later define all of the particular chord lengths proportionally.

$$Re\sqrt{C_L} = \frac{1}{\mu}\sqrt{\frac{2\rho W}{AR}} \tag{1}$$

For the two configurations of *Eternity* ($66\,Wh$ and $132\,Wh$), calculated mean $Re\sqrt{C_L}$ are 90k and 110k respectively. After obtaining the working regime, priority is given to the $((C_l^{1.5}/C_d)_{max})$ value of the airfoil, as the endurance performance is the main objective for the design. SD7037 airfoil is selected as a reference because it is widely known and used in soaring competition gliders. By using

[2] xfoil.mit.edu

http://www.imavs.org/

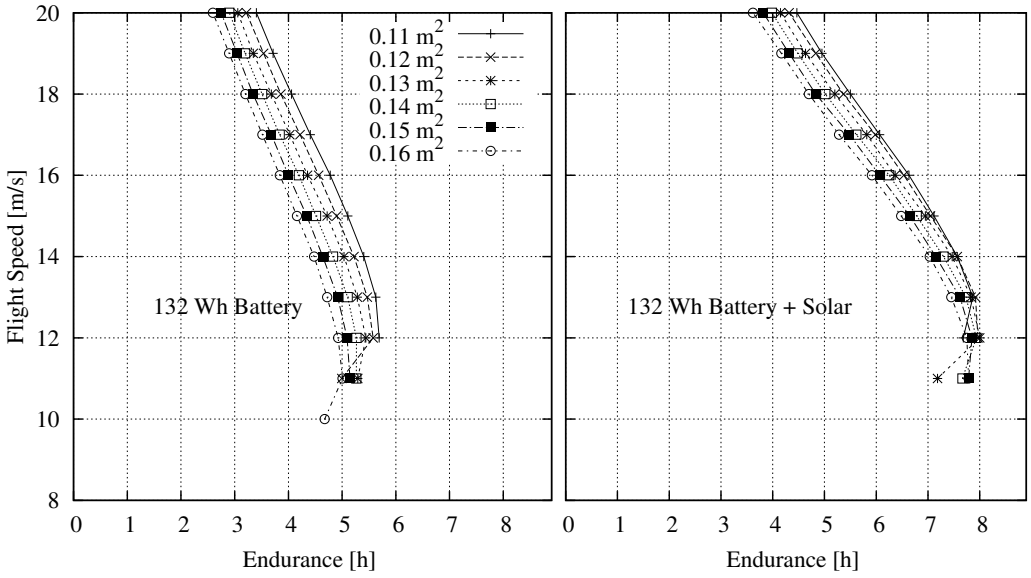


Figure 7: The effect of various flight speed and wing area on the endurance performance of non-solar and solar configurations with 132Wh of on-board energy.

the QDES routine of XFOIL, the laminar run on the top surface is extended by increasing the aft-loading the airfoil. The transition ramp slope is slightly reduced and the pressure distribution at the bottom end is increased. This resulted an increase on the pitching moment increase on the final airfoil ($\Delta C_m = 0.025$), which was acceptable as the conventional configuration was selected. The suction peak is smoothed, resulting in a better continuity of the flow and postponement of the transition. It is known that the bubble loss contributes significantly to the form drag on the low-Reynolds airfoils. One way of reducing the bubble size is to move it farther upstream by forcing the transition earlier. However, it will not be beneficial to completely prevent the occurrence of the bubble, because the laminar flow will be shortened dramatically. There is an optimum position for the bubble placement and size that corresponds to the minimum drag, and in low Reynolds airfoil design the control of the bubble becomes more critical compared to minimizing skin friction [5].

In order to improve the airfoil performance along the span and decrease the viscous drag caused by inappropriate airfoil location, a series of airfoils are designed by referencing PKMB500.

PKMB803 is designed particularly for its own working regime, which is $Re\sqrt{C_L} = 30000$. Figure 8 shows the placement of the designed airfoils along the span, with a drag comparison of each with the root airfoil PKMB500 in their corresponding flow regime.

Having a smooth surface on the wing while transitioning from PKMB500 to PKMB803 on an elliptical planform created a dominant restriction on the thicknesses of PKMB601 and PKMB702. As the chord distribution has already been fixed, adding the root and the tip airfoil thickness values automatically defines the thickness distribution. With this thickness restriction, the only way found to slightly improve the performance of the middle airfoils was to compromise from the maximum lift coefficient and concentrate on the cruise coefficient regime corresponding to best endurance, which is around $C_L = 0.9 - 1.0$.

http://www.imavs.org/

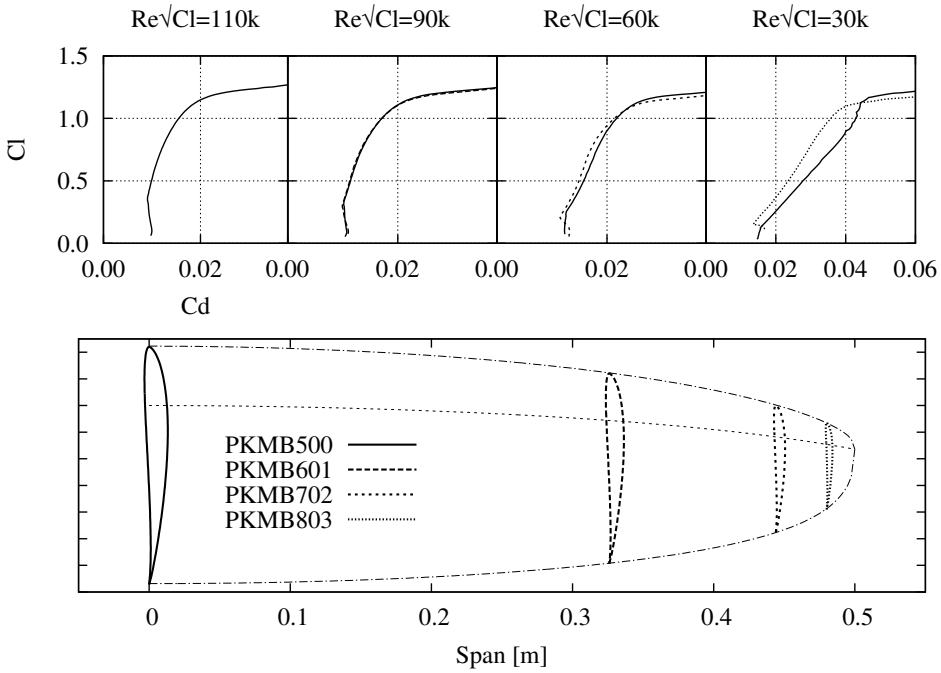


Figure 8: Placement of each airfoil is shown along the half span. The upper graphs shows the C_l vs C_d plots of each airfoils at their corresponding working regimes with the comparison to the root PKMB500 airfoil. The planform shown is given as demonstration and the real planform is shown later in figure 9.

Final dimensions of the two *Eternity* versions are shown in the figure 9, with their specifications in table 10.

6 Windtunnel Tests

In order to obtain the aerodynamic characteristics of the aircraft in a controlled environment, a wind-tunnel campaign was conducted at ISAE *S4* wind-tunnel. An internal six component force and moment balance was used for the measurements.

Instead of manufacturing a separate wind-tunnel model, a fully equipped ready-to-fly version was manufactured and used for the tests. This was initiated in order to save some time, as one particular model takes one person about 3-4 weeks to build, including all the necessary equipment integration. Figure 11 shows the mounting of the aircraft in the wind-tunnel. The control surfaces and the motor of the aircraft were controlled by the on-board Paparazzi autopilot, which was connected to a computer via serial connection.

The experiments are conducted for different speeds in order to see the performance of the aircraft at each flight speed and evaluate the aerodynamic characteristics for a wider flight range. For each wind-tunnel speed the aircraft pitched from -6 to 12 degrees in order to obtain the performance

http://www.imavs.org/

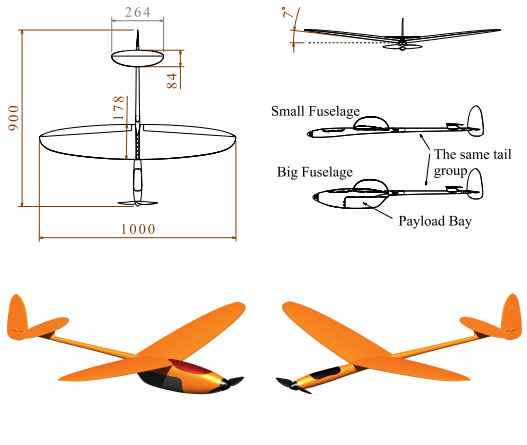


Figure 9: *Eternity* design with two different fuselages.

		Small	Big
Wing Area	$[m^2]$	0.145	0.145
Total Mass	$[kg]$	0.85	1.35
Length	$[m]$	0.9	0.87
Wing AR	$[-]$	6.89	6.89
Horizontal tail AR	$[-]$	4.0	4.0
Vertical tail AR	$[-]$	2.0	2.0
Horizontal tail volume	$[-]$	0.4	0.4
Horizontal tail arm	$[m]$	0.45	0.45
Vertical tail volume	$[-]$	0.03	0.03
Vertical tail arm	$[m]$	0.53	0.53

Figure 10: Geometrical specifications of the two *Eternity* configurations.

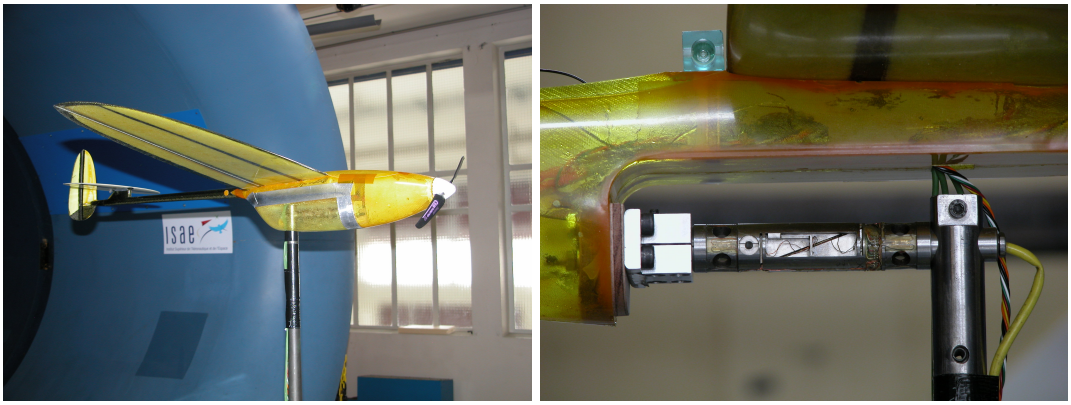


Figure 11: The mounting of Eternity on S4 Wind-tunnel, the payload compartment of the Eternity has been used for the mounting of the internal balance, and then the payload bay covered in order to protect the balance and to have the accurate fuselage shape.

polars.

The expected influence of the Reynolds number is clearly visible in figure 12; as the flight speed decreases, both the range (C_L/C_D) and the endurance ($C_L{}^{1.5}/C_D$) performance decrease. Attention should be given to the fact that figure 12 does not present the equilibrium condition performance. In order to have the equilibrium plot, elevator deflection has to be taken into account as well as at which it corresponds to zero pitching moment.

According to the previous analyses, the best endurance performance is achieved at $C_L = 0.9-1.0$, where the cruise speed corresponds to $12.5-13.0\,m/s$, which is really close to stall speed of the big configuration that is $11.5\,m/s$. In real-life applications, for mini-micro UAV scales that are generally flying around $10-25\,m/s$, sustaining airspeed accurately is really hard when the small momentum of the aircraft and the big ratio of the wind gusts to the flight speed are considered. Therefore, increasing

http://www.imavs.org/

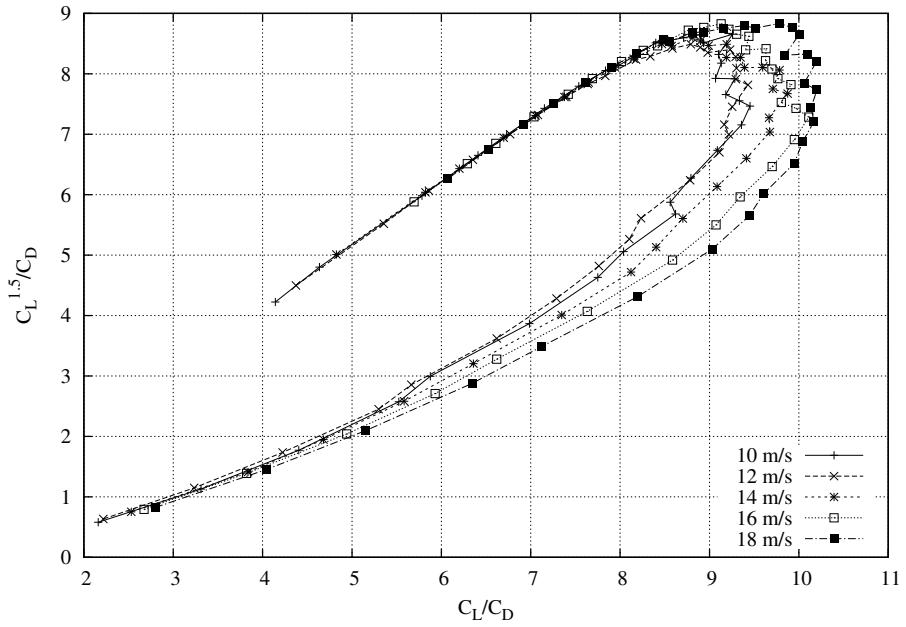


Figure 12: Effect of flight speed on the range and endurance performance characteristics.

the flight speed results in a reasonable safety margin. In the end of this decision, the cruise speed is selected as $14\,m/s$.

7 Flight Tests

Flight tests have been conducted by DelairTech[3] UAV company under a partnership contract between ENAC, ISAE and TTT[4] for the MIDLE project.

Close to $10\,hours$ of flights have been recorded. The first prototype that has also been used in the wind-tunnel tests is given to the company for flights. Only one battery has been used, nevertheless the total weight of the aircraft (in flight ready state) was $930\,g$, just a bit heavier than the expected single battery configuration by $100\,g$. Additionally, the big fuselage is used as the thin fuselage was not build at that time. The longest flight that has been achieved was $2\,hours$ and $7\,minutes$ by using $54\,Wh$ of energy. The battery has given only 81% of its expected energy. The maximum glide ratio (L/D) and the power coefficient $({C_L}^{1.5}/C_D)$ is found to be 13.9 and 13.0 from the glide tests.

Considering all the differences of the test configuration compared to the design configurations, the flight test results act as a reference point rather than a direct comparison.

[3] www.delair-tech.com
[4] www.toulouse-tech-transfer.com

http://www.imavs.org/

8 Overall Design Conclusion

The main objective of this study was to show the feasibility of the *Long Endurance Mini UAV* concept. *Eternity* shows a great perspective about what can be achieved with a one-meter aircraft. Finally, it can be concluded that with a **one-meter** aircraft that fits into an even smaller half meter carry-on luggage, **4 hours** of flight are possible, and this can be enhanced up to **8 hours** on a sunny day with the help of solar cells.

		Small	Big
Cruise Speed	$[m/s]$	11.5	14.0
Cruise Lift Coefficient	$[-]$	0.71	0.76
Stall Speed	$[m/s]$	9.2	11.6
Max Bank Angle at V_{cruise}	$[deg]$	50.0	46.0
Turn Radius at Max Bank	$[m]$	11.3	19.3
Battery capacity	$[Wh]$	62.5	125
Payload Mass	$[kg]$	0.05	0.10
Total Mass	$[kg]$	0.85	1.35
Solar Conditions			
Endurance	$[h]$	7.48	6.2
Range	$[km]$	309	312
Non-Solar Conditions			
Endurance	$[h]$	3.3	3.92
Range	$[km]$	138	197

Table 3: Expected performance of the two *Eternity* configurations.

References

[1] Sion power, the rechargeable battery company.

[2] Arreté du 21 mars 2007 relatif aux aéronefs non habités qui evoluent en vue directe de leurs opérateurs, May 2010.

[3] Murat Bronz. Experimental comparison of 1-meter conventional, pusher and tractor flying wing aircraft configurations in s4 wind-tunnel. Technical report, ISAE, 2009.

[4] Murat Bronz. *A Contribution to the Design of Long-Endurance Mini Aerial Vehicles*. PhD thesis, l'Ecole National de l'Aviation Civile, 2012.

[5] Mark Drela. Low-reynolds number airfoil design for the mit daedalus prototype: A case study. *Journal of Aircraft*, 25(8):724–732, August 1988.

[6] Mark Drela. An analysis and design system for low reynolds number airfoils. In University of Notre Dame, editor, *Conference on Low Reynolds Number Airfoil Aerodynamics*, June 1989.

[7] Mark Drela and Harold Youngren. Athena vortex lattice, Sep 2004.

[8] Dr.-Ing. S. F. HOERNER. *Fluid-Dynamic Drag*. 1965.

[9] Daniel P. Raymer. *Aircraft Design : A Conceptual Approach*. American Institute of Aeronautics and Astronautics,Inc., Virginia,VA, 2006.

[10] Jan Roskam. *Airplane Design*. DAR Corporation, Lawrence, Kansas, 2000.

http://www.imavs.org/

Rotary Wing Micro Air Vehicle Endurance

Klaus-Peter Neitzke*

University of Applied Science Nordhausen, Nordhausen, Germany
neitzke@fh-nordhausen.de

Abstract

One of the first questions to pilots of rotor based electric driven micro air vehicles is the possible flight time of the system. The answer is not easy sometimes. The possible flight time is influenced by a lot of important details of the micro air vehicle and the flight mission. At the end all components will influence the result. This will start with the battery, continues with the electronic equipment, the mechanical design and ends with the motors and the propellers. Finally the weather condition and the flight mission will influence the maximal possible flight time. This paper starts with a theoretically investigation of the flight performance. The results will be compared with several flight test data from two different rotor based electric driven micro air vehicles. These two vehicles were used during the International Micro Air Vehicle Conferences IMAV 2011 and 2012. At the end this paper will give a battery recommendation for different missions at different weather conditions.

http://www.imavs.org/

Contents

1 Symbols **2**

2 Used test vehicles - quadrocopter "Wanze" and "Ninja" **2**

3 Battery - the source of the energy **3**

4 Propeller - there the thrust is coming from **5**

5 Flight time - this time the air vehicle can stay in the air **5**

6 Optimal battery mass - more flight time is not possible **8**

7 Selection of the battery - the optimal battery for different flight missions **8**

7.1 Mission A: Hobby flights without special needs 8

7.2 Mission B: Pylon races or speed or climb competitions 8

7.3 Mission C: Long range flights - fly the maximum possible time 9

7.4 Mission D: Stable flights - steady flight under windy weather conditions 9

8 Outlook **9**

*Professor for Automation Systems, University of Applied Science Nordhausen, Germany

http://www.imavs.org/

1 Symbols

E	=	energy content in [J]
U	=	voltage in [V]
I	=	current in [A]
T	=	time in [s]
m	=	mass in [kg]
d	=	rotor diameter in [m]
g	=	acceleration of gravity = $[9.81 \frac{m}{s^2}]$
h	=	altidute in [m]
v	=	flow velocity in $[\frac{m}{s}]$
ρ	=	density of the air = $[1.25 \frac{kg}{m^3}]$
A	=	sum of the rotor area in $[m^2]$
D	=	specific power of the battery in $[\frac{J}{kg}]$
η	=	efficiency
N	=	number of rotors or battery cells

2 Used test vehicles - quadrocopter "Wanze" and "Ninja"

The two quadrocopter Q-240 "Wanze" and Q-500 "Ninja" were used to validate the theory of this investigation. They are convenient for this comparison because they are similar from the technical point of view. But they have different dimensions. The "Wanze" dimension is half size of the "Ninja" dimension. Both vehicles are an in-house development oft the author. The two vehicles were used during the International Micro Air Vehicle Conferences IMAV 2011 and IMAV 2012, see figure 1.



Figure 1: The quadrocopter "Wanze" and "Ninja" used at IMAV 2011 and IMAV 2012

The air frame based on a sandwich structure made from carbon plates and balsa wood. The wood between the carbon plates is 5 mm thick end grain. The result is a very stiff and light design. The complete electronic closed loop control will be performed by a microprocessor ATMEGA328 on a printed circuit board with the dimension of 20 x 30 mm. The three gyroscope sensors ADXRS610 measure the rotation speed of the vehicle. The three acceleration components will be measured by the sensors ADXL322. All sensors are implemented in a 17 mm red cube, see figure 1 . This cube was used in different other projects, see references [4], [3] and [5]. The software is an in-house development as well. The basis is the Arduino platform,

the size of the code is 8 Kbytes about. Part of the code is the closed loop control including the rotary matrix. The closed loop frequency is 400 Hz. The RC receiver is a single satellite from a standard Spectrum receiver. The receiver is connected at the serial port of the controller board. The standard electronic speed controllers receive typical pulse wide modulation information 1.1 ms to 1.9 ms from the controller. For the "Wanze" AirAce 3-blade left/right propellers were used, the diameter is 100 mm. The motors are Hacker A10/12S powered by electronic speed controller Hype 6 Ampere. The basic weight of the quadrocopter is 119 gram. The dimensions of the "Ninja" are bigger than the "Wanze" dimensions. Scaling factor is two. The "Ninja" is using GWS 8x4 3-blade left/right propellers. The diameter is 200 mm. The propellers are powered by Axi motors 2808/34 and electronic speed controller Hype 10 Ampere. The basic weight of the "Ninja" is 352 gram. Both quadrocopter were used to validate the theory described in the next chapters.

3 Battery - the source of the energy

Lithium polimer batteries (Li-Poly) were used in the area of model aircraft presently. The main parameters are mass and specific power of the battery. The energy content can be calculated from the voltage, the current and the discharge time, see equation (1).

$$E = U \cdot I \cdot T \tag{1}$$

To have an impression of the energy content of a lithium polimer batteries we can do a thought experiment. The energy content from equation (1) can be used to lift the battery only. The idea is to convert this electrical energy to mechanical energy. This energy can be calculated from equation (2). The energy will be calculated from the mass, the acceleration of gravity and the lift altitude.

$$E = m \cdot g \cdot h \tag{2}$$

In the thought experiment we suppose the electrical energy will be transformed to the mechanical energy completely. We assume there is no loss of energy and the mass of the motor and the wire is negligible. The set up maybe looks like figure 2.

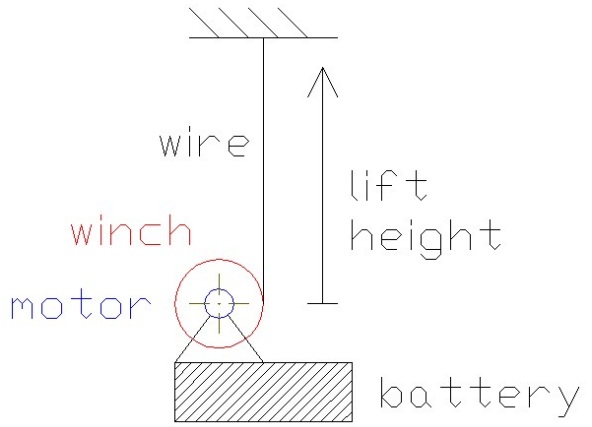


Figure 2: The set up of the battery thought experiment

http://www.imavs.org/

The electrical energy from the battery will be transformed to lift energy completely. The energy from equation (1) and equation (2) is identical, so the lifting altitude can be calculated from equation (3).

$$h = \frac{U \cdot I \cdot T}{m \cdot g} \tag{3}$$

Example: Lithium polimer batteries with 3 cells and a capacity of 1 500 mAh has a mass of 142 gram. He is able to lift his own weight in the thought experiment up to 43 kilometer see equation (4). In the reality this value will be smaller because of losses in all components. But the value of 43 kilometer is very high, he shows how much energy is installed in a lithium polimer battery.

$$h = \frac{11.1 \cdot 1.5 \cdot 3600}{0.142 \cdot 9.81} \text{ meter} \approx 43 \text{ kilometer}. \tag{4}$$

The fact that this value is not dependent from the size of the battery is remarkable. A heavier battery will have more electrical energy. The quotient of energy and mass is similar for one type of battery. It is possible to calculate a specific power of the battery. This quotient shows how much energy is installed in one kilogram of the battery. In table 1 several lithium polimer batteries are listed. They are sorted by the number of the cells, the mass and the capacity.

number of cells	voltage/[V]	mass/[g]	capacity/[mAh]	energiy density/[J/kg]
2	7,4	28	350	333000
2	7,4	20	350	466200
2	7,4	27	450	444000
2	7,4	30	500	444000
2	7,4	43	740	458456
2	7,4	41	800	519805
2	7,4	68	1200	470118
3	11,1	41	450	438585
3	11,1	78	910	466200
3	11,1	142	1500	422113
3	11,1	165	1800	435927
3	11,1	329	3600	437252
			average value=	444638

Table 1: Overview on several lithium polimer batteries

For further calculations we suppose an average specific power of the battery D of 444 kJ/kg. Of course always there will be examples with a higher or smaller specific power, but the value 444 KJ/Kg is a good approximate value. With this value we can estimate the mass of several lithium polimer batteries, see equation (5).

$$m_{battery} = \frac{E}{D} \tag{5}$$

A lithium polimer battery with 3 cells and a capacity of 10 000 mAh will have a mass of 0.9 kg for example. This calculation we need later, when we select an optimal battery for several missions and quadrocopter applications.

http://www.imavs.org/

http://www.imavs.org/

4 Propeller - there the thrust is coming from

A good overview on the aerodynamic of propellers is given in the chapter 4 of the reference [1]. The air above the propeller goes to the propeller and creates a jet below the propeller, see figure 3.

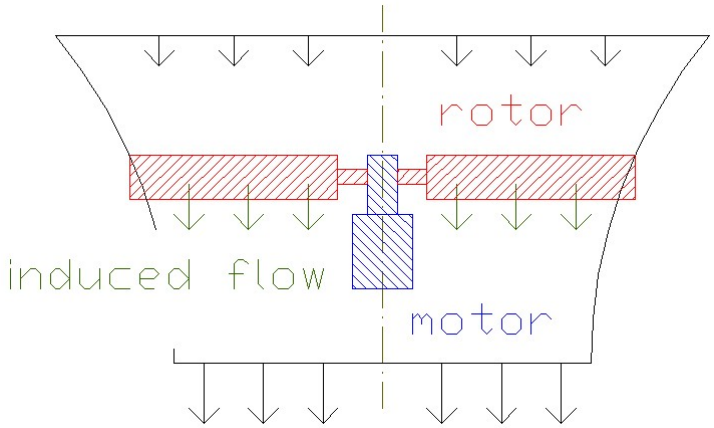


Figure 3: The induced flow velocity near to the propeller plane

In the propeller plane there is an induced velocity. This velocity can be calculated from equation (6).

$$v = \sqrt{\frac{m \cdot g}{2 \cdot \rho \cdot A}} \tag{6}$$

The induced velocity is high, when the air vehicle has a high weight and a small rotor area. To calculate the installed thrust power we have to multiply the induced velocity by the thrust. As result we have an equation to calculate the necessary power for the hover flight, see equation (7).

$$P = \frac{(m \cdot g)^{1.5}}{\sqrt{2 \cdot \rho \cdot A}} \tag{7}$$

The power is dependent from density of the air, the acceleration of gravity the mass of the vehicle and the rotor area. A heavy quadrocopter with small propellers needs more power. To fly with a small amount of power it is necessary to use big propellers on a quadrocopter with a low mass. It is possible to decrease the amount of energy by using the ground effect. The level of the induced velocity is limited because of blockage and this limits the value of power.

5 Flight time - this time the air vehicle can stay in the air

The flight time can be calculated by the energy content and the necessary power of the air vehicle, see equation (8).

$$T = \frac{E}{P} \tag{8}$$

No we use equation (5) and (7) in the equation (8). And we assume the flight mass is the sum of the empty weight of the air vehicle and the mass of the battery. Now we have an equation for the flight time, see equation (9).

$$T = \frac{E \cdot \sqrt{2 \cdot \rho \cdot A}}{(g \cdot (m_{empty} + m_{battery}))^{1.5}} = \frac{E \cdot \sqrt{2 \cdot \rho \cdot A}}{(g \cdot (m_{empty} + \frac{E}{D}))^{1.5}} = \frac{m_{battery} \cdot D \cdot \sqrt{2 \cdot \rho \cdot A}}{(g \cdot (m_{empty} + m_{battery}))^{1.5}} \tag{9}$$

To select a battery for a flight mission we can use equation (9) for the calculation of the possible flight time. A high value for the energy content and a large propeller area will produce long flight times. More difficult is the situation for the mass of the battery. This value is included in the numerator and the denominator of the equation (9). In the next chapter we will see there is a optimal battery mass to reach a maximum for the flight time in equation (9).

Two characteristic curves were shown in figure 6 for the quadrocopter "Wanze" and "Ninja". Important is the fact the real flight time will be smaller then the calculation from equation (9). This theoretical value we have to multiply be an efficiency of the system. This effect is caused by losses in the battery, electronic, motors and propellers. We add all this losses in one efficiency factor for the complete system, see equation (10).

$$T_{effective} = \eta \cdot T \tag{10}$$

The efficiency factor we can determine in flight tests. The size of the battery was varied, see figure 4 and figure 5 .

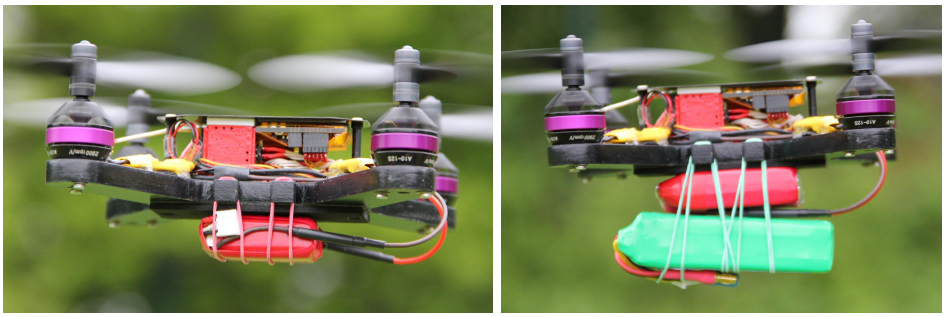


Figure 4: Comparison of different batteries for the quadrocopter "Wanze"

We compare the theoretical flight time versus the real flight time. The quotient is the efficiency factor. This efficiency is dependent on the components of the air vehicle. To simplify the handling of the equation (9) we can assume a voltage of one cell of the lithium polimer battery of 3.7 Volt. The density of the air will have a value of 1.25 $\frac{kg}{m^3}$. The acceleration of gravity is 9.81 $\frac{m}{s^2}$. These information are implemented in the new factor 0.32, see equation (11). With this modification we will have an practical equation to estimate the possible flight time in minutes.

$$\frac{T_{effective}}{[minutes]} = \eta \cdot 0.32 \cdot \frac{\sqrt{N_{rotors}} \cdot \frac{d_{rotor}}{[mm]} \cdot N_{cells} \cdot \frac{capacity}{[mAh]}}{(\frac{m_{empty}}{[gram]} + \frac{m_{battery}}{[gram]})^{1.5}} \tag{11}$$

http://www.imavs.org/



Figure 5: Comparison of different batteries for the quadrocopter "Ninja"

In figure 6. comparison between the calculated and the tested flight times are shown. The results fits well if we consider an efficiency factor for the quadrocopter "Wanze" of 0.19. For the quadrocopter "Ninja" this value is 0.35. The lower value for the "Wanze" based on the smaller propellers. The small dimension of the propeller will have addition aerodynamic losses. Since the same electronic components were used in both quadrocopter the constant losses of these components we decrease more the global efficiency of the smaller quadrocopter "Wanze".

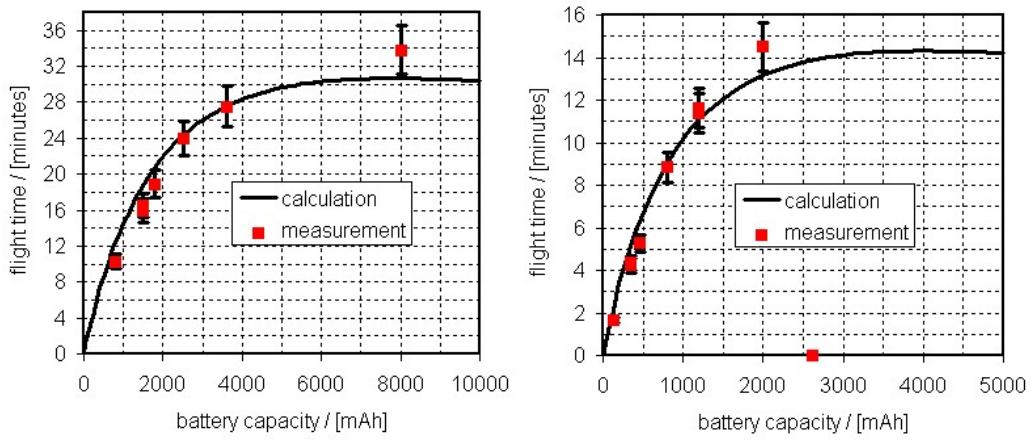


Figure 6: Comparison calculated versus measured flight time of the "Wanze" and "Ninja"

The "Wanze" was not able to lift the battery with the capacity of 2600 mAh. On the left hand side of figure 6 the flight time was quoted with 0 minutes. For practical applications is has to be ensure the air vehicle is able to lift lift the battery. This is an additional limitation in the selection of the batteries for a mission. The comparison of the flight times for the "Wanze" and the "Ninja" shows the measured flight times for higher capacities of the battery is higher than the calculated values. Reason for this is the increase of efficiency for higher thrust levels. The aerodynamic condition increase with the increase of flow velocity for higher lift levels, based on a higher Reynolds-Number.

http://www.imavs.org/

6 Optimal battery mass - more flight time is not possible

In figure 6 we see there is a optimal flight time. If we increase the size of the battery there is no longer flight time possible. This optimum can be found by the investigation the quotient (12) in equation (9).

$$\frac{m_{battery}}{(m_{empty} + m_{battery})^{1.5}} \tag{12}$$

The flight time has the maximal value when the mass of the battery is twice as high then the mass of the empty air vehicle. This is an easy result, see equation (13).

$$m_{battery} = 2 \cdot m_{empty} \tag{13}$$

This result is visible in figure 6. The result is not depended from the size of the air vehicle. It is not important which type of battery is used. Different type for example nickel cadmium batteries will have different flight times but the optimum is still reached when the battery mass is twice as high then the mass of the empty air vehicle. For the quadrocopter "Wanze" the value for the maximal flight time would be 3967 mAh. But this set up was not possible. The "Wanze" was not able to lift this battery. For the "Ninja" the optimal value was 7822 mAh. This value was tested and validated in the flight tests, see figure 6 .

7 Selection of the battery - the optimal battery for different flight missions

For the selection of the battery several point have to be investigated. First the number of cells will be adjusted. For this point the rotor and motor specification is important. A good selection for the quadrocopter "Wanze" was 2 cells and for the "Ninja" 3 cells. This delivers enough thrust and is a good combination for the motor and the rotor speed. To optimise the battery for the flight mission it is necessary to distinguish between different flight missions. Typical missions A, B, C and D are possible:

7.1 Mission A: Hobby flights without special needs

To enjoy in flying with the air vehicle a small battery is a good selection. This limits the costs for the battery and decreases the crash mass of the vehicle. The mechanical loading of the frame will not have a high value in a crash situation. The battery mass should be 50 % of the mass of the empty air vehicle. This is a good estimation for the first flights. The quadrocopter "Wanze" uses for this mission a battery with 2 cells and a capacity of 800 mAh. The possible flight time is 9 minutes. The quadrocopter "Ninja" will use a 3 cells battery with the capacity of 1500 mAh battery and reaches a flight time up to 16 minutes.

7.2 Mission B: Pylon races or speed or climb competitions

There we need power and a low mass of the air vehicle. Maybe a different number of cells is profitable for this mission. But at the end the best solution is the smallest battery, who is able to fulfil the mission. Maybe the necessary flight time is 1 to 2 minutes only. Important is the fact, that the needed power is higher for this mission than for the calculated needed power for hovering flight. Experimental results expect an increase of power for this mission by 40 % see references [2]. The battery capacity has to be enlarge by this value.

http://www.imavs.org/

7.3 Mission C: Long range flights - fly the maximum possible time

For this mission the mass of the battery mass should be twice as high then the mass of the empty air vehicle. With this battery set up we reach the maximum flight time. There is no need to use bigger batteries, the flight time will be smaller. If we select a small battery the flight time will shorter. But if we use a battery with the same mass then the empty air vehicle the possible flight time is 92 % of maximal possible flight time. Maybe there is no need to reach the last 8 % by doubling the battery mass and cost. This is related to more noise and a higher crash mass of the air vehicle.

7.4 Mission D: Stable flights - steady flight under windy weather conditions

There we need a higer mass of the quadrocopter. With a high mass of the air vehicle we reach a high induced flow velocity near to the rotor plane, see equation (6). If there is an additional gust based from the weather turbulence, then will be a change of the incoming flow at the propeller blade. The local wind spee and the angle of attack can be different from the reference flow condition. This will increase or descries the local lift at the propeller blade element. This will end up with additional oscillations of the air vehicle. We have to avoid this for flights with cameras an other payload. Often there is the need for a stable flight. The disturbance of the reference flow condition is bigger for smaller induced flow velocities in the propeller plane. If there is a high level of induced flow velocity available, then an additional gust will have less influence on this reference flow condition, see figure 7.

The flow condition on a propeller blade element is shown in the upper part of the figure 7. The incoming flow on the blade is the sum of the induced flow and the peripheral speed of the propeller. If there is an additional gust, see lower part of the figure 7, then the incoming flow on the propeller will be different. The change in the angle of attack is bigger for lower velocities, see left hand side of the lower part of figure 7. If there is a high velocity of the induced flow and the peripheral speed of the propeller then a turbulence gust is not able to disturb this flow condition. The change in the angle of attack is smaller then for the condition with lower speed. Since the angle of attack is bigger, the flow conditions are less stable. This will induce more oscillations of the vehicle and there is the need to have more compensation from the closed loop control of the air vehicle.

The higher flow velocities we will achieve by a high mass of the flight vehicle. There is no mandatory need to increase this high mass with the battery only. It can be used an additional mass or a heavier structure mass of the airframe to. If the quadrocopter is able to lift this additional mass, then this is a good set up for this mission. It can be used a mass higher than for mission C but the flight time will be smaller. At the end the flight will be more stable.

For all Missions is important to know, there are more flight limitations. For very small batteries in mission B it is necessary to check if the batteries are capable to deliver the electrical current for the mission. For heavy batteries the first check should be the comparison of the maximum available lift from air vehicle. For all mission buffers in the capacity of the batteries should be included.

8 Outlook

There will be more powerful batteries in the future. More specific power of the battery will be available. This will increase the maximal possible flight time. To reach long flight times today

http://www.imavs.org/

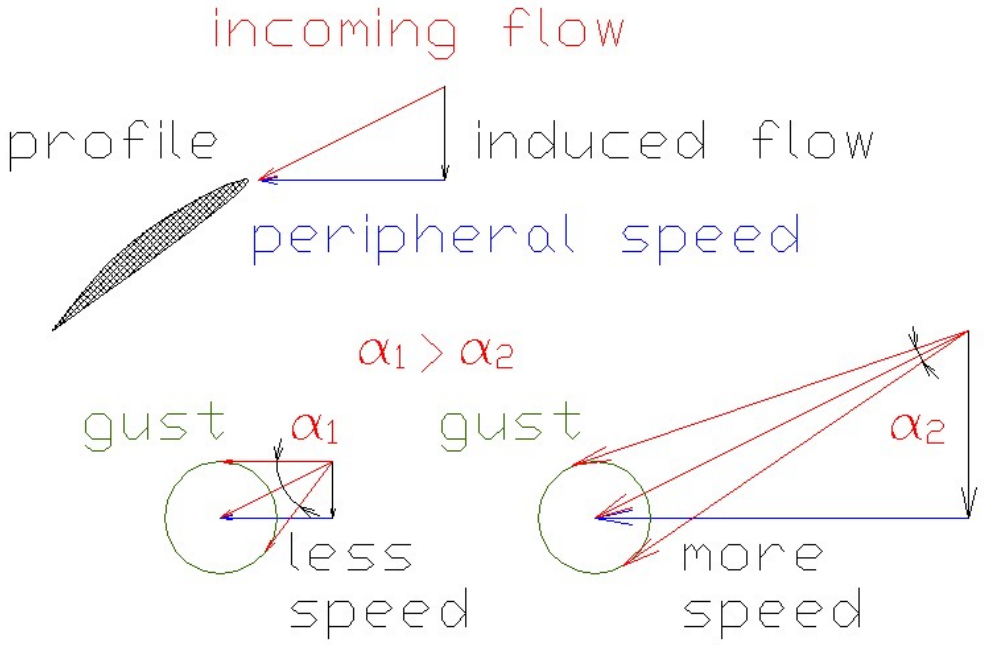


Figure 7: Flow conditions on a rotor blade element for low and higher flow speeds

there is the need to select all components of the air vehicle carefully. Flights in a low level turbulence condition will increase the flight time. Additional the ground effect can increase the maximal possible flight time. But for this environments there are no interesting missions available unfortunately. An other topic is to use more batteries for longer missions. They can be dropped stepwise after discharging. To estimate the flight time of rotor based electric driven micro air vehicles there is the need to investigate a lot of details of the vehicle. The other solution is to perform flight experiments. The best solution is to have the right balance between theoretical investigations and experimental flight tests.

References

[1] Walter Bittner. *Flugmechanik der Hubschrauber*. Springer Verlag. ISBN 978-3-540-88971-7.

[2] Mikrokopter community. Flight time calculator for quadrocopter in the forum mikrokopter. http://forum.mikrokopter.de/topic-21052.html.

[3] Klaus-Peter Neitzke. Autopilot based on an aircraft with 490 mm wing span the forum diydrones. http://diydrones.com/forum/topics/presentation-30-x-40-mm.

[4] Klaus-Peter Neitzke. The quadcopter "zecke q185" in the forum diydrones. http://diydrones.com/forum/topics/new-quadcopter-zecke-q185-max.

[5] William Thielicke. Multirotor blogspot from william thielicke. http://shrediquette.blogspot.de.

http://www.imavs.org/

Adaptive proprotors as applied to convertible MAVs

Peng Lv[*], Fazila Mohd-Zawawi[†], Emmanuel Benard[‡], Sebastien Prothin[§], Jean-Marc Moschetta[¶] and Joseph Morlier[‖]

[1] Université de Toulouse; ISAE, DAEP;
10 av. Edouard Belin - BP 54032 - 31055 Toulouse Cedex 4, France
peng.lv@isae.fr

[2] Université de Toulouse; Institut Clément Ader; ISAE, UPS, EMAC, INSA;
10 av. Edouard Belin - BP 54032 - 31055 Toulouse Cedex 4, France
fazila.mohd-zawawi@isae.fr

http://www.imavs.org/

Abstract

A passive twist control is considered as an adaptive way to maximize the overall efficiency of a proprotor developed for convertible Micro Air Vehicles (MAV). Incorporated into a database of airfoil characteristics, Blade Element Momentum Theory (BEMT) is implemented to predict the performance of proprotors at low Reynolds numbers. Using this model, it is found that low twist allows for efficient hovering while high twist helps to forward flight. The Centrifugal Force Induced Twist (CFIT) concept is proposed to realize the required torsion of proprotor between hover and forward flight. Tip mass is used to provide the nose-down twisting moment by centrifugal force and stabilize the flexible blade. Classical Lamination Theory (CLT) is employed to estimate the torsion behavior of glass/epoxy laminate blade and to study the feasibility of CFIT concept. The results indicate that the predicted torsion of CFIT blade is of the same level with required deformation. The laminate blades were tested in hover and forward flight modes, with deformations measured by Laser Displacement Sensor (LDS). In rotor mode, the laminate blade can generate approximately -9° torsion at blade tip rotating at 1,300 RPM. By contrast, at 800 RPM and inflow velocity 8m/s, it is capable of providing around -5° torsion at blade tip in propeller mode.

1 Introduction

Convertible rotor aircraft has been developed for versatility services for several years, as it combines the merits of a helicopter and and airplane. It has the capacity of Vertical/Short TakeOff and Landing (VSTOL) like a helicopter, through tilting rotor, then converts to forward flight at relatively high speed, as an airplane. The three main convertible rotor configurations are the tilt-rotor, tilt-wing and tilt-body. Tilt-rotor aircraft was developed in early 1950s, exemplified by the Bell XV-3 operating the first transition from hover to forward flight. In 1970s, XV-15 demonstrated the feasibility of tilt-rotor concept. The success of XV-15 led to the V-22 project, leading to the first production tilt-rotor aircraft in the world. Recently, the tilt concept has attracted the attention of MAV research

[*] PhD student, DAEP, ISAE
[†] PhD student, DAEP/DMSM, ISAE
[‡] Associate Professor, DAEP, ISAE
[§] Researcher, DAEP, ISAE
[¶] Professor, DAEP, ISAE
[‖] Professor, DMSM, ISAE

http://www.imavs.org/

community. In 2008, Shkarayev and Moschetta introduced the concept of tilt-body MAV, which has a tilt-body configuration and is capable of flying in hover and forward flight [1]. This configuration of tilt-body MAV which was successfully tested in flight.

Designing a proprotor to operate efficiently in hover and forward flight presents a challenge since the inflow velocity and thrust requirement for each flight condition are quite distinct. In hover, the inflow velocity is small and the proprotor must provide high thrust to support the aircraft weight. By contrast, in forward flight, the inflow velocity is relatively large and the low thrust must only overcome the drag. The difference in the inflow and thrust requirements between the two flight modes suggests different blade twist and chord distributions. In terms of twist effect on efficiency, high blade twist on the proprotor allows the aircraft to fly faster and more efficiently, whereas low blade twist increases the efficiency in hover. In 1983, McVeigh obtained the twist of XV-15 proprotor through linear interpolation of twist between ideal rotor and propeller by a compromise analysis [2]. Although this trade-off solution provided an acceptable performance on XV-15, the stiff proprotor with certain twist cannot maximize the efficiency for both flights. In 1988, Nixon proposed a passive blade twist control for the proprotor on XV-15 [3]. The study demonstrated successfully the feasibility of the passive blade control on conventional tilt-rotor aircraft. The small proprotors also suffer the problem caused by different twist between hover and forward flight. However, due to the small size of MAV, the complex tailored cross section of blade for passive twist control based on conventional tiltrotor aircraft is not available any more. Therefore, based on CFIT concept, composite laminate is proposed to be a more practical method for proprotor blade of MAV.

A key issue to study flexible blade is to use validated predictive simulations and therefore, in the domain of aeroelasticity, to measure accurately deformations. Optical measurement techniques have been developing for some years in aerodynamics, materials and structure, such as Holographic Interferometry (HI), Electronic Speckle Pattern Interferometry (ESPI), Projection Moiré Interferometry (PMI) and Digital Image Correlation (DIC) [4]. In 1998, Fleming obtained the 3-D deformation of rotor blade using PMI technique [5]. However, it has low sensitivity for in-plane deformation and moderate for out-of-plane deformation. By contrast, DIC has a relatively high sensitivity that can reach 1/30,000 of the test field [6]. In 2011, Lawson demonstrated the deformation of a rotating blade using DIC [7]. The technique was found to have many advantages including high resolution results, non-intrusive measurement, and good accuracy over a range of scales. However, DIC needs a preprocessing which is to apply a stochastic speckle pattern to the surface by spraying it with a high-contrast and non-reflective paint. This complex painting will probably affect the stiffness of small MAV blades. Hence, in this study, LDS based method was developed to measure blade deformation in both of hover and forward flight.

2 Optimum twist of MAVion proprotor

MAVion, a tilt-body MAV developed by ISAE, is designed to fly in both hover and forward flight, as shown in Fig. 1. The typical wing Reynolds number of its operation conditions is below 100,000. In hover mode, the nominal thrust coefficient is 0.1 while it becomes 0.03 in forward flight. The detailed operation conditions are shown in Tab. 1.

To obtain the optimum twist distributions of proprotor in hover and forward flight, the specifications of MAVion proprotor are presented in Tab .2 as arbitrary constraints. Flat plate cross section was used as airfoil of flexible blades since it is relatively practical for CFIT concept study in terms of fabrication. In addition, thin plates, cambered or flat, with extremely thin leading edge, exhibit characteristics less sensitive to variations in Reynolds number and turbulence[8]. These airfoils achieved better lift-to-drag ratios than conventional airfoils at low Reynolds numbers. On one hand,



Figure 1: Wind tunnel model of MAVion.

	Hover	Forward flight
RPM	1500	1200
Velocity (m/s)	0	10
Advance ratio, J	/	1.25
Thrust (N)	2	0.3
Thrust coefficient	0.01	0.003

Table 1: Operation conditions of MAVion

the thickness of flat plate should be thin enough for aerodynamic performance. On the other hand, it is necessary to consider the stacking thickness of potential adaptive proprotor based on composite materials. Hence, flat plate with thickness 2.5% was selected as the airfoil for proprotor. Five linear built-in twists of proprotor are defined first, i.e. -10°, -15°, -20°, -25° and -30° blade tip twist relative to root. Based on airfoil characteristics from XFOIL [9], BEMT was used to compute airloads and

Number of blades	2
Stacking axis of airfoils	0.25 on the chord
Pitch axis	0.25 on the chord
Airfoils	Flat plate
Chord	0.03m
Radius	0.2m
Hub radius	0.03m

Table 2: MAVion proprotor specifications

to analyze the efficiencies in rotor and propeller modes. It has been implemented and validated in a previous study [10]. The optimum twist distributions for hover and forward flight is found when the efficiencies of rotor and propeller can be maximized, respectively.

In rotor mode, after the inflow ratio λ is determined, the induced power coefficient C_{P_i}, the profile power coefficient C_{P_0} and the hovering efficiency FM can also be calculated using standard approach

http://www.imavs.org/

[11]:

$$C_{P_i} = \int_{r=0}^{r=1} 4\lambda(r)^3 r dr, \tag{1}$$

$$C_{P_0} = \frac{\sigma}{2} \int_{r=0}^{r=1} C_d(r) r^3 dr, \tag{2}$$

$$\mathrm{FM} = \frac{C_{P_{ideal}}}{C_{P_{meas}}} = \frac{C_T^{3/2}/\sqrt{2}}{C_{P_i} + C_{P_0}}. \tag{3}$$

Where r is the non-dimensional radius, dr is the non-dimensional length of each element, σ is the local solidity, C_d is the local drag coefficient and C_T is the thrust coefficient.

In propeller mode, the propulsive efficiency η is defined by conventional trust and power coefficients:

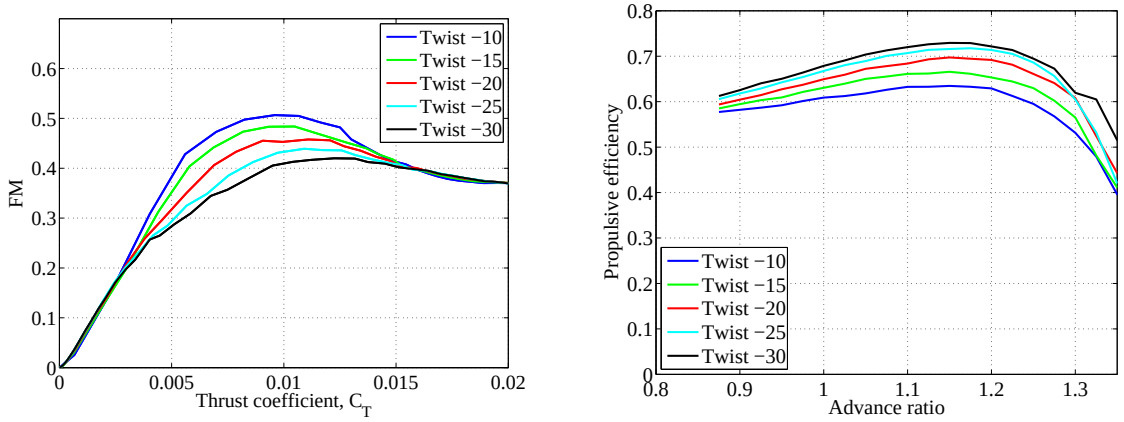
$$C_T = (\pi^3/4)\sigma C_y r^3 F^2/[(F + \sigma K')cos(\phi)]^2, \tag{4}$$

$$C_P = C_T \pi r C_x / C_y. \tag{5}$$

$$\eta = C_T J / C_P, \tag{6}$$

where J is the advance ratio, C_x and C_y are the propeller force coefficients, F is Prandtl's tip-loss factor, K' is equivalent interference factor and ϕ is the inflow angle. Propeller performance is typically described by plots of C_T, C_P and η vs J.

Fig. 2(a) shows the twist effect on hovering efficiency - Figure of Merit (FM) - with the variation to thrust coefficient. Thrust coefficient is adjusted by collective pitch for the five blades with built-in twist -10°, -15°, -20°, -25° and -30°, respectively. The blade with built-in twist -10° has the maximum efficiency at C_T=0.01. In Fig. 2(b), the propulsive efficiencies of the five blades are all analyzed for a fixed value C_T=0.003. They vary with the advance ratios in forward flight. The blade with twist -30° exhibits the highest efficiency of all, at advance ratio J=1.25. Overall, high twist of blade is beneficial for forward flight while low twist can improve hovering efficiency. According to the mission requirement of MAVion, the solution of proprotor for MAVion is given in Tab. 3.



(a) Twist effect on rotor efficiency

(b) Twist effect on propeller efficiency

Figure 2: Twist effect to small-scaled proprotor

http://www.imavs.org/

	Hover	Forward flight
Built-in twist (degree)	-10	-30
Collective pitch (degree)	31	54

Table 3: Optimum proprotor of MAVion

3 Concept of adaptive proprotor

CFIT concept aims at realizing a deformable, durable and stable blade for MAV proprotor, which are characterized by the deformation, failure performance and dynamics behavior. The initial step of the procedure is to select suitable reinforcing fiber for the laminate blade. The significant factor to select a reinforcing fiber for small-scale proprotor is linked to its tailoring capacity. Glass/epoxy was determined for the MAV flexible proprotor in current study. In order to improve the aerodynamic performance of flat plate at low Reynolds number, the thickness of laminate blade should be thin enough to provide a low thickness-to-chord ratio. Thus, two typical laminate configurations are considered here, a symmetric laminate $[\theta\ \theta]_{\mathrm{T}}$, and a antisymmetric laminate $[\theta - \theta]_{\mathrm{T}}$. A critical issue in the design of a laminate blade is failure analysis. The comparison of strength of UD and angle-ply laminates was described based on a typical carbon/epoxy composite (AS4/3501-6) using first ply failure analysis [12]. All of the uniaxial tensile strength, uniaxial compressive strength and in-plane shear strength of the angle-ply laminate are evidently higher than those of the off-axis UD material. Thus, in current study, antisymmetric laminates $[\theta\ -\theta]_{\mathrm{T}}$ are employed as balanced laminates. The layup of specimen in the current study is selected as $[45 - 45]_{\mathrm{T}}$.

The Classical Lamination Theory (CLT) demonstrates that, for a general composite laminate, the forces and moments on it are related to the strains and curvatures at reference surface. The 6×6 matrix consisting of the components A_{ij}, B_{ij}, and D_{ij} $(i, j = 1, 2, 6)$ is the laminate stiffness matrix, is also called ABD matrix. In order to be able to obtain the strains and curvatures at the reference surface in terms of the force and moment resultants, the ABD matrix is inversed and then becomes the laminate compliance matrix consisting of the components a_{ij}, b_{ij}, and b_{ij}. The torsion can be directly given by the curvature:

$$\kappa^0_{xy} = b_{16}N_x + b_{26}N_y + b_{66}N_{xy} + d_{16}M_x + d_{26}M_y + d_{66}M_{xy} \tag{7}$$

Where the N_x, N_y and N_{xy} are the tensile forces in the directions of x, y and xy. Besides, M_x, M_y and M_{xy} represent the moments in varied directions. Terms b_{16}, b_{26}, b_{66}, d_{16}, d_{26} and d_{66} are the corresponding compliance factors.

The adaptive proprotor should not only be flexible to produce the required deformation, but also stable to maintain the required twist in each of flight mode. For a stable blade, the blade CG must be ahead of the aerodynamic center [13]. Hence, in CFIT flexible concept, tip mass is designed to adjust the global CG of laminate blade. In addition, tip mass is beneficial for improving the nose-down twisting moment and increasing the torsion of laminate blades. On one hand, tip mass is required to be heavy enough to be able to adjust the global CG. On the other hand, it should be light enough to provide weight efficiency for the application on MAVs. Tab. 4 shows the basic parameters on the blade with tip mass. In order to estimate the twist of laminate blades, the tensile force and twisting moment of a spinning blade are introduced to CLT model. At any local section of blade, the total centrifugal force and total nose-down moment are defined by [13]:

$$N_x(y) = F_{cf}(y) = \int_y^R \Omega^2 m_y y dy + m_T \Omega^2 R, \tag{8}$$

http://www.imavs.org/

Table 4: Basic parameters of blade

Weight of blade	7.0g
Weight of tip mass	6.5g
Chord	30mm
Length	185mm
Radius	200mm
Thickness of blade	0.75mm
Length of tip rod	35.0mm
Diameter of tip rod	5.0mm
Diameter of tube	7.0mm

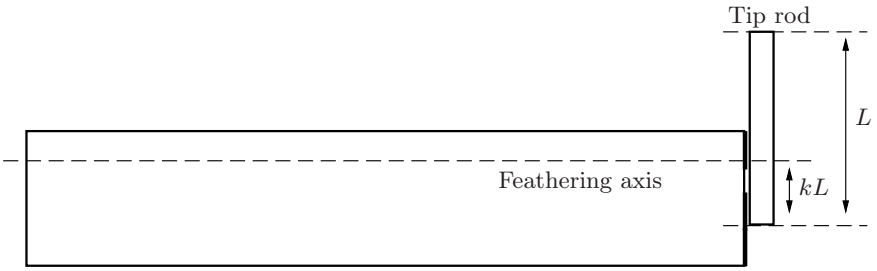


Figure 3: Definition of rotating axis factor k

$$M_{xy}(y) = M_{nd}(y) = I_\theta \Omega^2 (R - y) sin(\theta) + I_T \Omega^2 sin(\theta_T). \tag{9}$$

Where x is the chordwise coordinate, y is the spanwise coordinate, R is the proprotor radius, Ω is the rotation speed, m_y is the local mass of laminate blade and m_T is the tip mass. In addition, θ is the local twist angle and θ_T represents the twist angle at the blade tip. The torsional moment of inertia of the blade I_θ is small compared to the torsional moment of inertia of the tip rod I_T, thus the nose-down moment acting on the blade airfoil is negligible compared to the nose-down moment acting on the tip mass. Similarly, the centrifugal force on the blade airfoil is negligible compared to the one on the tip mass. Hence,

$$F_{cf}(y) = m_T \Omega^2 R, \tag{10}$$

$$M_{nd}(y) = I_T \Omega^2 sin(\theta_T). \tag{11}$$

The moment of inertia of the tip rod I_T is defined as:

$$I_T = \int_{-kL}^{(1-k)L} \rho x^2 s dx. \tag{12}$$

Where L is the length of tip rod and k is the rotating axis factor. k is used to define the distance from one side of tip rod to the twisting axis, as shown in Fig. 3. For example, k=0.5 means that the twisting axis is on the midpoint of tip rod. If k is between 0 and 1, the twisting axis is on the tip rod. Else, the tip rod rotates about the off-body axis. Changing density ρ with mass m, section area s and rod length L, it becomes:

$$I_T = \int_{-kL}^{(1-k)L} \frac{m}{sL} x^2 s dx = \frac{m}{3L}[((1-k)L)^3 - ((-k)L)^3]. \tag{13}$$

http://www.imavs.org/

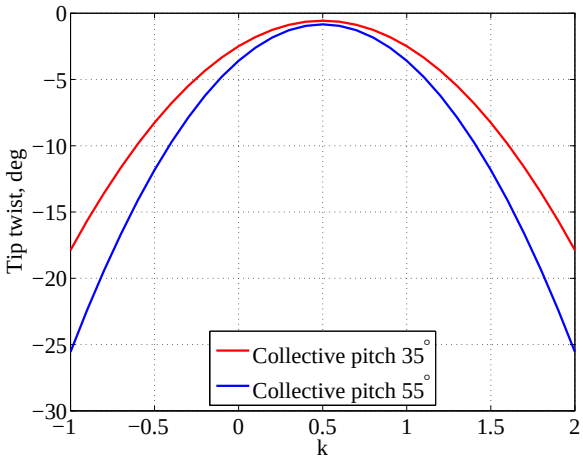


Figure 4: Predicted torsion of adaptive blade

Here, the twist behavior of laminate blades with low and high collective pitches based on 1,500 RPM. As discussed above, low collective pitch requires low twist for rotor mode while high twist is needed at high collective pitch for propeller mode. As it can be seen in Fig. 4, at both low and high collective pitches, the tip torsion shows the symmetrical behavior with respect to k=0.5 since the torsional moment of inertia of tip rod I_T at k=0.5 is minimum. Through sliding the tip rod, the CFIT concept blade can achieve the beneficial torsion at the required level.

The final laminate blades with tip masses are shown in Fig. 5. The bending and torsion distributions during rotation are considered as the blade deformation, since beneficial torsion can increase the overall performance while bending tends to decrease the thrust.

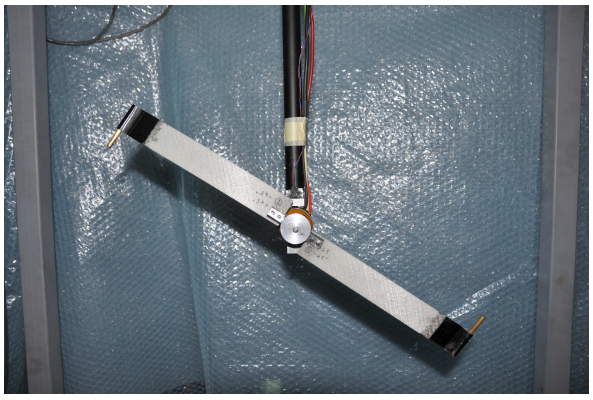


Figure 5: Adaptive proprotor

http://www.imavs.org/

http://www.imavs.org/

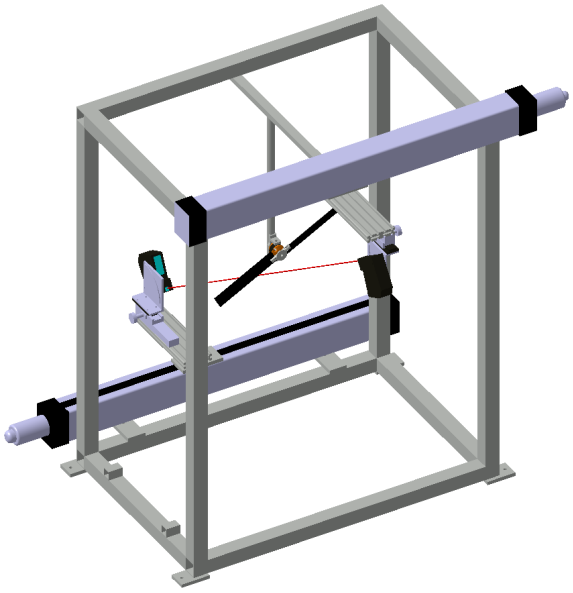
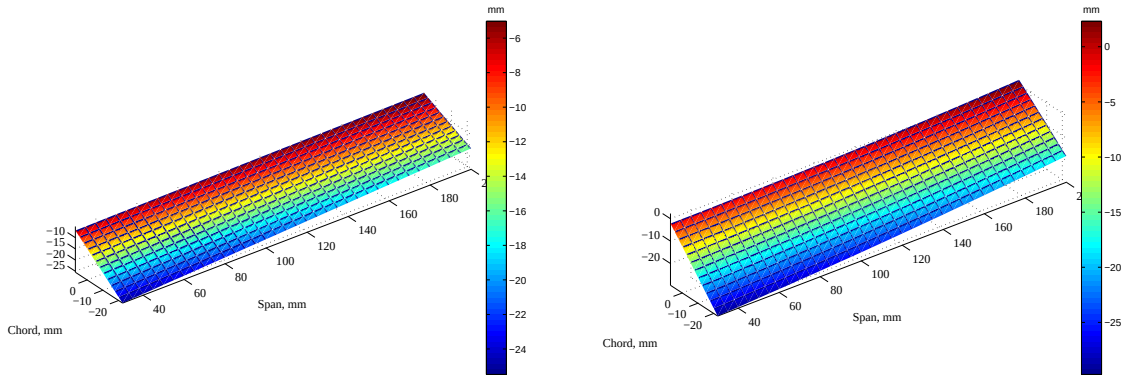


Figure 6: LDS rig model

4 Testing and discussion

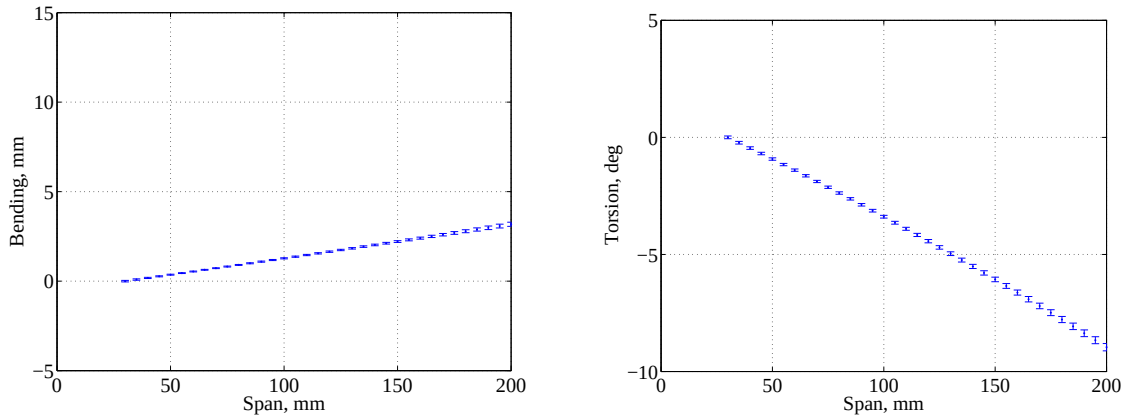
In order to measure the deformation of rotating laminate blade, a LDS system was developed, as shown in Fig. 6. To avoid any disturbance on flow field, the two LDSs were fixed at an incidence angle to measure the blade deformation. The LDS used in this experiment is a KEYENCE LK-G502. The sampling frequency was set to 10,000Hz. Diffuse reflection mode of LDS was used for measurements. The distance of reference was 500mm, and the range of measuring was between -250mm to 500mm. For a long range measurement, the measuring accuracy is typically ±0.1% of Full Scale (FS). In



(a) Rotor mode: RPM 1,300

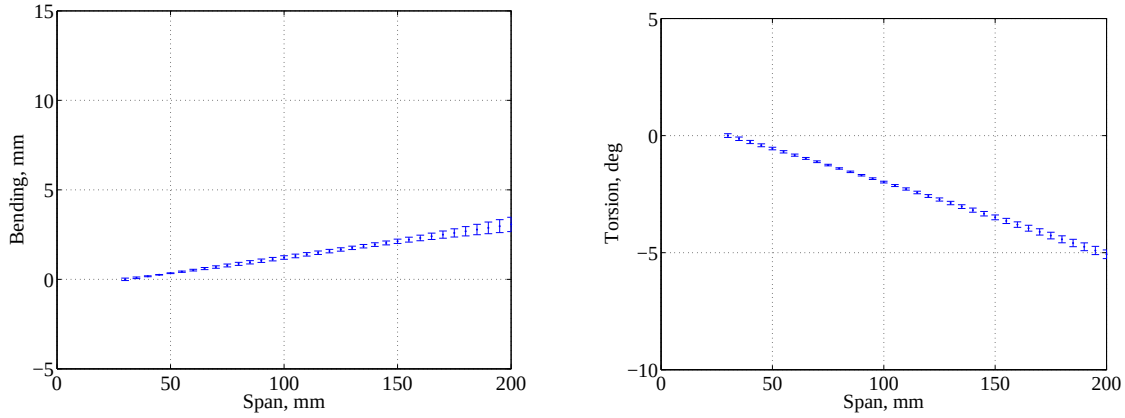
(b) Propeller mode: RPM 800 and inflow 8m/s

Figure 7: Shape reconstructions of adaptive proprotor



(a) Bending (b) Torsion

Figure 8: Rotor mode



(a) Bending (b) Torsion

Figure 9: Propeller mode

order to reconstruct the deformed blade and to extract the bending and torsion, the post-processing methodology based on polynomial surface fitting was developed for LDS technology, including error propagation of measurement based on the Kline-McClintock method [14].

The results on shape reconstruction are shown in Fig. 7. In both hover and forward flight, the obvious deformation occurred during rotation. The detailed bending and torsion were extracted according to surface function. From Figs. 8 and 9, it was found that the CFIT blade is capable of generating -8.96° torsion at RPM 1,300 in hover mode. In propeller mode, with high collective pitch, it generated around -5° torsion at RPM 800, inflow 8m/s. Meanwhile, the maximum values of bending at blade tip are less that 5mm for both modes. In propeller mode, the blade can afford inflow and keep away from large bending.

http://www.imavs.org/

5 Conclusion

The BEMT results showed high twist of blade is suitable for propulsive efficiency while low twist is beneficial for hovering efficiency. The CFIT concept of adaptive proprotor was verified first by CLT model. Glass/epoxy was used for the adaptive proprotor with the antisymmetric stacking sequence. A post-processing methodology was proposed to reconstruct deformed blade based on LDS technique, including uncertainty analysis based on Kline-McClintock method. In both of hover and forward flight, the adaptive proprotor produced evident torsion. However, the torsion generated in propeller mode was not capable of reaching the required twist. In coming tests, the motor of high torque capacity is expected to be used for high RPM and torsion. The adaptive proprotor based on CFIT concept showed the application on MAVion. The developed models and LDS technique are reliable tools for designing and analyzing the proprotor made of composite material.

References

[1] S. Shkarayev, J. M. Moschetta and B. Bataille, *Aerodynamic Design of Micro Air Vehicles for Vertical Flight* (Journal of Aircraft, Vol. 45, No. 5, 2008) pp. 1715-1724.

[2] M. A. McVeigh, H. J. Rosenstein and F. J. McHugh, *Aerodynamic Design of the XV-15 Advanced Composite Tilt Rotor Blade* (39th Annual Forum of the American Helicopter Society, St. Louis, MO, May 1983).

[3] M. W. Nixon, *Improvements to Tilt Rotor Performance through Passive Blade Twist Control* (NASA Technical Memorandum 100583, April 1988).

[4] S. Rajpal, *Optical Methods of Measurement: Whole-field Techniques* (Second edition, Francis and Taylor/CRC Press, 2009).

[5] G. A. Fleming and S. Gorton, *Measurement of Rotorcraft Blade Deformation Using Projection Moiré Interferometry* (Proceedings of the 3rd International Conference on Vibration Measurements by Laser Techniques: Advances and Applications, SPIE the International Society for Optical Engineering, Ancona, Italy, June 1998) pp. 514-527.

[6] T. Schmidt and J. Tyson, *Full-Field Dynamic Displacement and Strain Measurement Using Advanced 3D Image Correlation Photogrammetry: Part 1* (Experimental Techniques, Vol. 27, No. 3, 2003) pp. 47-50.

[7] M. S. Lawson and J. Sirohi, *Measurement of Deformation of Rotating Blades Using Digital Image Correlation* (Proceedings of 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Denver, Colorado, April 2011).

[8] B. J. Hein and I. Chopra, *Hover Performance of a Micro Air Vehicle: Rotors at Low Reynolds Number* (International Specialists Meeting Unmanned Rotorcraft: Design, Control ans Testing, Chandler, AZ, January 2005).

[9] M. Drela, *XFOIL 6.9 User Guide* (Massachusetts Institute of Technology, 2001).

[10] P. Lv, S. Prothin, F. M. Zawawi, E. Benard, Joseph Morlier and J. M. Moschetta, *Study of A Flexible Blade for Optimized Proprotor* (ERCOFTAC International Symposium, Unsteady Separation in Fluid-Structure Interaction, Mykonos, Greece, June 2013).

[11] J. G. Leishman, *Principles of Helicopter Aerodynamics* (Cambridge Aerospace Series, 2nd edition, 2006).

[12] I. M. Daniel and O. Ishai, *Engineering Mechanics of Composite Materials* (Oxford University Press 1994).

[13] J. Sicard, *Investigation of an Extremely Flexible Stowable Rotor for Micro-helicopters* (Master's thesis, University of Texas at Austin, May 2011).

[14] S. J. Kline and F. A. McClintock, *Describing Uncertainties in Single-Sample Experiments* (Mechanical engineering, Vol. 75, January 1953) pp. 3-8.

http://www.imavs.org/

Scientific Session 2

Vision and image processing

Chairman : Guido De Croon

- Johannes Schellen, Christian Dernehl and Stefan Kowalewski. ***Optimizing Image Processing on OMAP3 with driver-level frame buffering and color space conversion***
- Guido De Croon, Hann Woei Ho, Christophe De Wagter, Erik-Jan van Kampen, Bart Remes and Qiping Chu. ***Optic-flow based slope estimation for autonomous landing***
- Laurie Bose and Arthur Richards. ***MAV Belief Space Planning In 3D Environments With Visual Bearing Observations***

http://www.imavs.org/

Optimizing Image Processing on OMAP3 with driver-level frame buffering and color space conversion

Johannes Schellen, Christian Dernehl and Stefan Kowalewski

RWTH Aachen University, Aachen, Germany
johannes.schellen@rwth-aachen.de
dernehl@embedded.rwth-aachen.de
kowalewski@embedded.rwth-aachen.de

http://www.imavs.org/

Abstract

Onboard image processing for unmanned aerial vehicles (UAVs) has become a popular method in the recent decades and the number of available hardware solutions has increased. Growing processing power and reduced weight and size of the embedded systems facilitate more computational power onboard the UAVs, making real-time image processing feasible. With respect to the software framework, the OpenCV library provides a set of useful functions to extract information from images. In this paper we first present a latency improvement over using OpenCV for camera input and show that the frame buffer optimization results in a latency reduction of up to one fifth compared to the OpenCV library. For the second part, we explain how utilizing direct driver access and hardware capabilities enables a faster color space conversion than OpenCV library functions. The color space conversion is tested with the L*a*b color space, which proves to be the right choice for our application, which is the detection of red objects in inhomogeneous light conditions. For our outdoor MAV application, the detection of six rectangular red objects takes no longer than about 50ms on average.

1 Introduction

An increasing computational power and decreasing prices have increased the applications for embedded systems and allow to run complex algorithms onboard. With the growing use of such systems, the research and development in the *UAV* (unmanned aerial vehicle) area has risen, so that a higher degree of automation becomes possible. Nowadays, the development in the UAV sector has advanced so far, that such vehicles can be bought in the retail store. However, the range of UAV types, which are sold off the shelf, is limited to either rotor-craft (tri-, quad-, hexa- and octacopter) or fixed wing vehicles and are controlled either with remote control or a fixed set of pre-programmed waypoints. More expensive systems include for example a first person view system, nevertheless, onboard video processing remains a niche in this sector, although in the research and development, there are more advanced systems available. One part of the video processing, which improves the degree of automation, is the segmentation, allowing to determine objects in the foreground from the background. Depending on the objects, which should be detected, the chosen method for the video segmentation is dependent on the application. For instance, objects can be identified by attributes, such as geometry, color or contrast. In this paper we present an optimization of a capture device for the Gumstix embedded on-chip processor[1], which is used in several research and development

[1]See http://www.gumstix.org .

projects[11][12][13]. Additionally our capture device allows to gather different image formats than RGB, that is L*a*b, for example. The L*a*b color space has been chosen for our application, which is to detect an arch, with six flags attached to its borders, which has been used in the IMAV 2012 competition[2]. Regarding the performance, the adapted implementation is compared against the *Open Computer Vision* (OpenCV) framework[3]. With respect to the target flight vehicle, the Embedded Systems Laboratory and the Institute for Flight Dynamics at the RWTH Aachen University have developed since 2011 a tilt-wing UAV, with hovering and fixed-wing capabilities[7][9], in which the video system is implemented.

The structure of the paper is given in the following. After the introduction into the topic has been given in this section, the related work on this field is briefly presented. Thereafter, an overview of the system is given and approaches to decrease the latency are explained. The results of the tests are evaluated in Section 4. Finally conclusions are drawn in the last section.

2 Related Work

In the 80's, Waxman et al.[17] developed a prototype for an vision system, capable of detecting roads from the bird perspective. About ten years later, Solka et al.[15] present similar approaches to detect landscape marks made by humans and vision-based landing becomes available[14].

With respect to our target computer system, the Gumstix series, which have been developed and extended in the last decade, the number of onboard processing applications has increased. Salazar et al.[13] port the GPS open source Toolkit (GPSTk) to the Gumstix series, making further GPS development easier and faster. Thinking of a solution with a Kalman-filter, AggieNav developed by Clavin Coopmans, offers an integrated system where the Kalman-filter is executed on the Gumstix hardware, which is connected to other subsystems, such as the inertial measurement unit[6]. Another Kalman filter approach to estimate the position via optical flow is explained in the paper from Kendoul et al.[10]. Eynard et al.[8] demonstrate a solution to approximate the altitude from a camera pointing downwards, while the complete position is calculated by Wang et al.[16] and Phang et al.[12], for instance. Finally, Lange et al. demonstrate how vision-based landing can be achieved[11].

3 System Design and Implementation

After the work of other research projects has been acknowledged, the system is described and the optimization procedure is developed. First the relevant hardware and software components are illustrated. Subsequently, the latency reduction is explained. Finally, the color space conversion to $L^*a^*b^*$ is presented.

3.1 The Hardware

The system on which the computer vision application is implemented consists of a *Gumstix Overo Computer-on-Module* with a 620MHz ARM CPU and 512MB of RAM, and a *Gumstix Caspa VL* camera. The camera uses an Aptina MT9V032 wide-VGA CMOS sensor. It is connected to the *Texas Instruments OMAP3 system-on-a-chip* (TI OMAP3 SoC) on the Gumstix Overo COM via a 28 pin flex cable which carries the parallel data signal and the I^2C configuration channel.

[2]See http://www.imav2012.org .
[3]See http://www.opencv.org .

http://www.imavs.org/

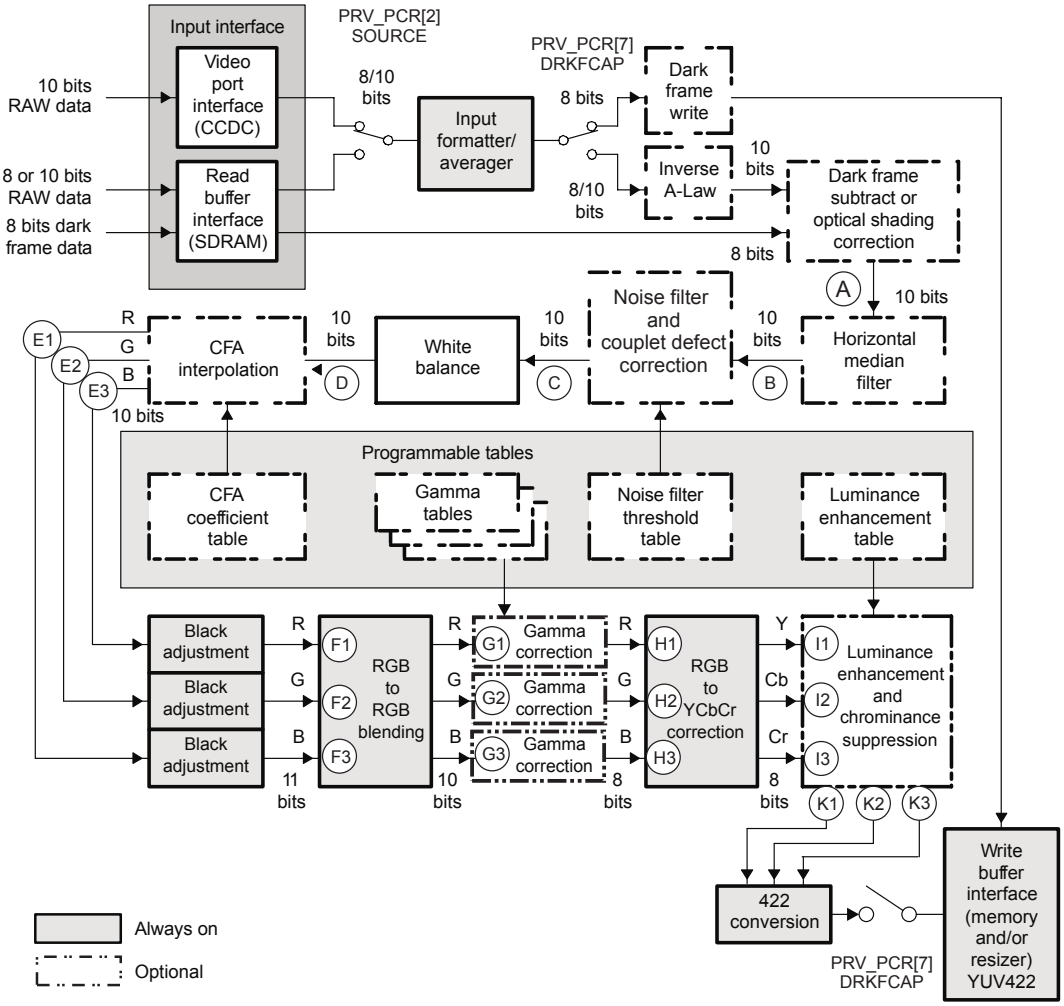


Figure 1: OMAP3 ISP Preview Engine Block Diagramm (taken from [1])

The TI OMAP3 SoC combines an ARM Cortex A8 CPU with a multitude of peripherals, such as a dedicated camera interface called *image signal processor* (OMAP3ISP). This peripheral receives sensor data from the Caspa VL camera, processes the pixels as they are read from the camera and uses DMA to write the images to main memory.

A central component of the OMAP3ISP is the *preview engine*, depicted in Figure 1, which takes the Bayer pattern sensor data, applies image enhancements and white balance scaling, interpolates the subpixel data and transforms the RGB pixels into YCbCr pixels. In Section 3.4 the reprogramming of the OMAP3ISP preview engine to output image data in the L*a*b* color space instead and the retrieving the data unmodified and with low latency is shown. First however, the implementation provided by the Open Computer Vision library is evaluated.

http://www.imavs.org/

3.2 OpenCV

The OpenCV library provides many image processing and computer vision algorithms as well as data structures and I/O functions. Applications can read camera images by using the OpenCV *VideoCapture* class, which takes one argument, being either the index of the camera or the name of a video file in case a recording is to be used instead of a live camera. Image data is converted from the native pixel format of the camera into the BGR pixel format which is used throughout OpenCV.

During the evaluation of the OpenCV VideoCapture class for the given application, a high latency between events in front of the camera and their appearance in images which are read through the VideoCapture class has been measured. The cause of this latency is in the OpenCV source code file `modules/highgui/src/cap_libv4l.cpp`. According to the comments at the beginning of the file, it is derived from the example code in the Video4Linux2 API documentation. Like the example code, `cap_libv4l.cpp` uses a FIFO queue of buffers. Four buffers are reserved for the queue by default, and when an image is transferred to the application, it is copied into yet another buffer.

3.3 Latency Reduction

A FIFO queue is useful for avoiding dropped frames if the application, on average, processes frame buffers faster than they are captured. On the other hand, a FIFO queue fills up and overflows if the processing time per frame is even just slightly longer on average than the time between two frame captures. In Figure 2, the effect of a full queue on the latency is depicted. The top half shows the utilization of four buffers in a FIFO configuration where the processing takes three and a half times as long as the frame time. The bars below visualize the latency between the time when an image is captured and the time when it is fully processed. The thick part of the bar shows the latency building up while the frame is in the queue, and the thin part represents the latency due to the actual processing by the application.

To avoid the high latency and the conversion into the BGR color space inflicted by the OpenCV VideoCapture class, a capture module which uses the Video4Linux2 API directly has been developed. This new module is based on the example code from the Video4Linux2 API documentation as well, but uses a *double buffering* strategy instead of a queue.

The V4L2 API uses two queues of memory mapped buffers: One is the *input queue* of empty buffers which are fed into the driver and the other is the *output queue* of buffers which the driver has filled with images from the camera. The capture module appends buffers to the input queue with the `VIDIOC_QBUF` IOCTL and dequeues buffers from the output queue with the `VIDIOC_DQBUF` IOCTL. A separate thread is utilized to dequeue buffers as soon as they become available. If the application hasn't retrieved a buffer from the capture module by the time the next buffer becomes available, this thread immediately moves the older buffer to the input queue. In this way the latest completed frame is always available to be retrieved by the application, while a second framebuffer is being filled by the camera driver. The lower half of Figure 2 visualizes the concept and shows the improved timing compared to a four buffer queue. Like the queue, double buffering achieves maximum throughput by avoiding delays. But unlike the queue, double buffering keeps the latency in the capture module below one frame time.

For the implementation of this concept, a total of four buffers are used: One filled buffer awaits retrieval by the application. The driver writes to the second buffer, and the application processes the image in the third buffer. Finally, the fourth buffer waits in the input queue to avoid underruns during the time when the next buffer has been filled by the driver and the separate thread has not yet returned the older completed buffer to the input queue. Buffers swap roles as necessary so that no image data needs to be copied in the process.

http://www.imavs.org/

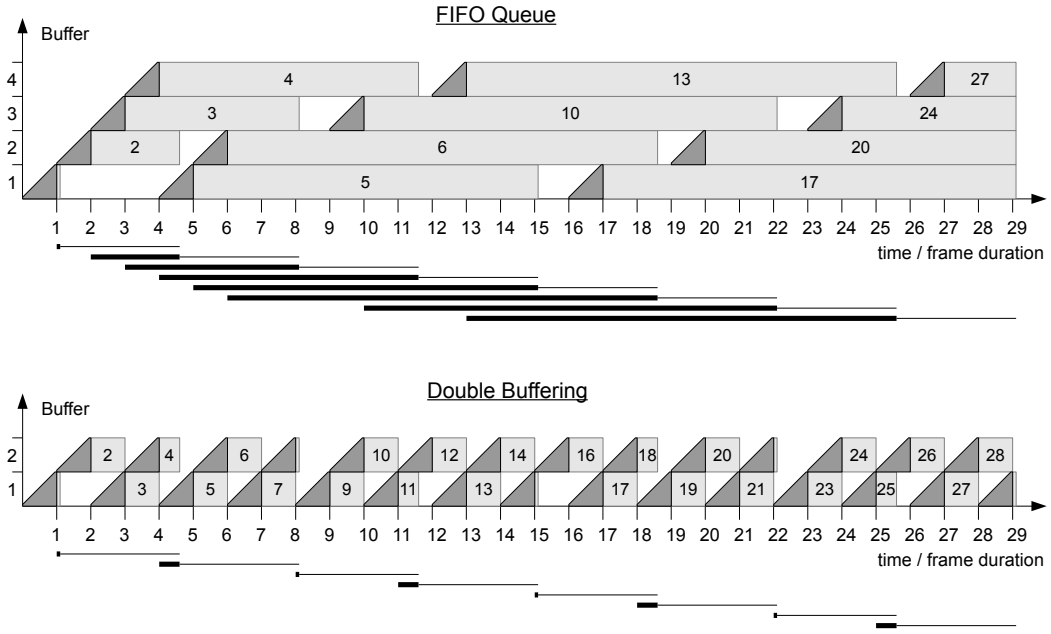


Figure 2: Latency of image capturing with a FIFO queue compared to double buffering

3.4 L*a*b* Color Space Conversion

With a low latency method for retrieving image data from the camera interface in place, focus is set on getting the data in the desired color space. The *CIE L*a*b** [4] color space is defined by a conversion from the *CIE XYZ* [2] color space. In addition to this conversion, more processing steps are necessary if the original data is in a color space other than XYZ. In order to convert from linear RGB data to L*a*b* data, the data is first converted to XYZ by a linear transformation. If the original data is non-linear RGB, then an inverse gamma correction is applied before the conversion to XYZ. Starting with YUV data prepends another matrix multiplication to get RGB. Reading camera images in the usual YUV format and then performing the necessary transformation steps on the CPU creates a high overhead. This overhead can be avoided by changing the parameters of the OMAP3ISP preview engine to produce L*a*b* image data instead of YCbCr data. This offloads the calculations from the CPU and frees processing time for later stages of the given computer vision application. Direct use of the Video4Linux2 API enables the reading of the output data of the OMAP3ISP preview engine without further conversions.

Figure 1 shows the hardware function blocks on which the conversion is performed: After the color filter array interpolation and black level adjustment, the linear RGB pixels are processed by the *RGB to RGB blending*, *Gamma correction* and *RGB to YCbCr conversion* function blocks. The gamma correction is implemented as a lookup into three tables of 1024 unsigned 8-bit values each. A 3x3 matrix multiplication and the addition of an offset vector are executed for each of the *RGB to RGB blending* and *RGB to YCbCr conversion* steps. These calculations are performed in fixed point arithmetic, and the matrix and vector components are limited to different ranges and resolutions. The detailed specification can be found in the literature [1].

The starting point for the conversion is linear RGB, and two matrix multiplications with a gamma

http://www.imavs.org/

conversion in between can be used to perform the necessary calculations. With this in mind, we take a closer look at the mathematical background.

Converting from linear RGB to L*a*b* is a two step procedure: First linear RGB is transformed to CIE XYZ and afterwards converted to CIE L*a*b*. For linear RGB data with the primary colors and the white point of the sRGB [5] color space, the conversion to CIE XYZ is performed by applying the linear transformation in Equation (1). The conversion to L*a*b* is defined by Equation (2). The tristimulus values X_n, Y_n and Z_n for the white point D65 [3] are given in Equation (3).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412391 & 0.357584 & 0.180481 \\ 0.212639 & 0.715169 & 0.072192 \\ 0.019331 & 0.119195 & 0.950532 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

$$\begin{aligned} L &= 116\ f(Y/Y_n) - 16 \\ a &= 500[f(X/X_n) - f(Y/Y_n)] \\ b &= 200[f(Y/Y_n) - f(Z/Z_n)] \end{aligned} \tag{2}$$

$$f(x) = \begin{cases} x^{1/3} & \text{if } x > (6/29)^3 \\ (841/108)x + 4/29 & \text{if } x \leq (6/29)^3 \end{cases}$$

$$\begin{bmatrix} X_n \\ Y_n \\ Z_n \end{bmatrix} = \begin{bmatrix} 0,950456 \\ 1 \\ 1,089058 \end{bmatrix} \tag{3}$$

To map this conversion to the hardware, the conversion is decomposed into two matrix multiplications and one gamma correction as expressed in Equation (4), where F is the component-wise application of f from Equation (2).

$$\begin{bmatrix} L \\ a \\ b \end{bmatrix} = \begin{bmatrix} -16 \\ 0 \\ 0 \end{bmatrix} + M_{Lab} F \left(M_{XYZ} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \right)$$

$$M_{Lab} = \begin{bmatrix} 0 & 116 & 0 \\ 500 & -500 & 0 \\ 0 & 200 & -200 \end{bmatrix} \tag{4}$$

$$M_{XYZ} = \begin{bmatrix} 0.412391/0.950456 & 0.357584/0.950456 & 0.180481/0.950456 \\ 0.212639 & 0.715169 & 0.072192 \\ 0.019331/1.089058 & 0.119195/1.089058 & 0.950532/1.089058 \end{bmatrix}$$

The L* component ranges from 0 to 100 and the a* and b* components are in the -127 to 127 interval. While the L*a*b* conversion is specified for an RGB input range of 0 to 1, here the input values are in the range from 0 to 255. Accordingly, a scaling factor of $\frac{1}{255}$ is applied to the M_{Lab} matrix.

The output consists of 8-bit unsigned integers. In order to use the available resolution, the L*a*b* values are mapped to the full 0 to 255 interval. For the a* and b* components in the Cb and Cr channels, the required addition of 127 is implicitly performed by the OMAP3ISP preview engine, so no offsets need to be programmed for the color channels. The lightness channel however needs further attention. The output range is expanded from the 0 to 100 interval, and since the OMAP3ISP preview engine does not support negative offsets in the final processing step, the L* component is approximated according to Equation (5).

$$L = \frac{255}{100} \cdot \frac{116}{255} \cdot f(Y) - 16 \approx \frac{255}{100} \cdot \frac{100}{255} \cdot f(Y) = f(Y) \tag{5}$$

http://www.imavs.org/

Consequently, the matrix to be programmed into the *RGB to YCbCr conversion* step of the OMAP3ISP preview engine is given in Equation (6).

$$M'_{Lab} = \begin{bmatrix} 0 & 1 & 0 \\ 1.960784 & -1.960784 & 0 \\ 0 & 0.784313 & -0.784313 \end{bmatrix} \tag{6}$$

The *RGB to RGB blending* step is programmed with the M_{XYZ} matrix, and the gamma tables are computed according to the f function from Equation (2), but scaled to an input range of 0 to 1023 and an output range of 0 to 255.

When these configuration parameters are programmed using the `VIDIOC_OMAP3ISP_PRV_CFG` private IOCTL, the preview engine outputs 4:2:2 sampled L*a*b* data with a rescaled and offset L* channel. The kernel V4L2 driver is unaware of the modified configuration and continues to treat the data like 4:2:2 sampled YCbCr data, so that is the format which the application has to request in order to receive the L*a*b* data.

http://www.imavs.org/

4 Evaluation

For evaluation, a comparison to the OpenCV framework is presented regarding two aspects: In the first part, the throughput performance is shown with respect to the target platform, which has been introduced in Section 3.1. Thereafter, the latency reduction is measured using another ARM based system.

In the target application, the autonomous navigation of a MAV is aided by location information computed from visually recognizing an arch which is marked with six red flags. The algorithm for which the processing time is measured and plotted in Figure 3 consists of thresholding, correcting lens distortion, computing the center of gravity and size of each connected component, searching for a set of connected components which represent the six flags and computing the pose based on four point correspondences. Peaks within the plot result from a high number of different red objects within the scene or the entire absence of the arch. Both are situations in which many combinations of connected components have to be considered and rejected. Nevertheless, in many situations the entire algorithm completes in 40 milliseconds or less. In comparison, the conversion of a video frame from BGR to L*a*b* using the OpenCV `cvtColor` function takes about 40ms on the same hardware, on top of the overhead created by the YUV to BGR conversion before the frame is delivered to the application.

For the latency measurements, a Seagate Dockstar with a Logitech C270 USB camera has been selected as test environment. In the Dockstar, the Marvell Kirkwood 88F6281 SoC contains an ARMv5TE core at 1.2GHz. The system is equipped with 128MB of RAM. A two-color front panel LED is attached to two GPIOs. For the latency test, the camera, capturing 30 frames per second, is pointed at the LED. The test program turns the LED on and off at random times and measures the time until the state change is detected in a camera image by reading a single pixel from the image and comparing it to a threshold value. To simulate processing, the test program inserts a defined delay between the capturing of the image and the reading of the LED state. Figure 4 shows the measured latencies of test runs with OpenCV on one hand and our capture module on the other hand, each with 100ms and 500ms processing delays.

The measured latency matches the expected latency from theoretical considerations. Note that our implementation still achieves almost the same average latency with 500ms processing time in the application as the OpenCV VideoCapture class when only 100ms of processing time are available to the application per frame.

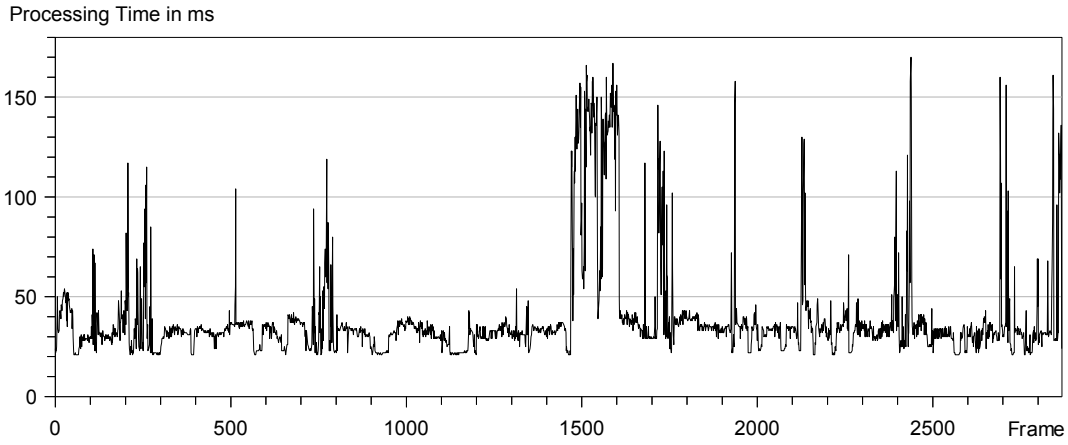


Figure 3: Measured processing time of the entire image recognition algorithm

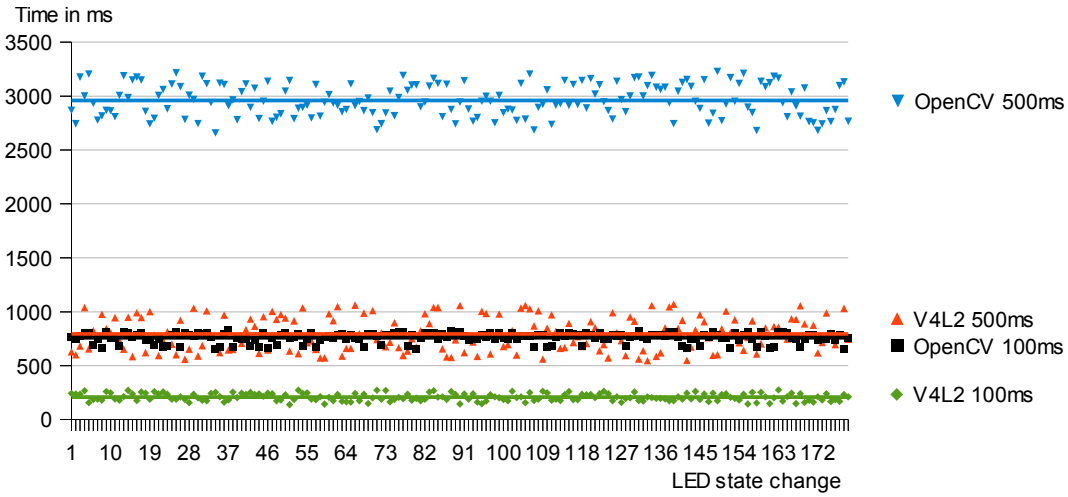


Figure 4: Measured latency from an LED state change to the detection in the camera image, with processing delays of 100ms and 500ms

5 Conclusion

In this paper we present an approach to improve the latency in comparison to the OpenCV implementation. Using double buffering instead of a FIFO queue allows the developer to implement more time critical applications. This improvement does not depend on properties of the Gumstix system and works on other hardware platforms as well.

Furthermore, we have demonstrated that utilizing the V4L2 API directly avoids the forced conversion to BGR and thereby opens up the possibility of moving the color space conversion to the OMAP3ISP preview engine. Compared to a software-only implementation with the OpenCV `cvtColor` library function, offloading the conversion to the dedicated hardware peripheral reduces the processing time by more than 40ms per frame. This step allowed us to double the frame rate of our

http://www.imavs.org/

application in many environments.

References

[1] *OMAP35x Applications Processor Technical Reference Manual*, 2012.

[2] CIE. Commission internationale de l'eclairage proceedings, 1931.

[3] CIE. 15-2004, colorimetry, 3rd edition, 2004.

[4] CIE. Iso 11664-4:2008 (cie s 014-4/e:2007), colorimetry, part 4: Cie 1976 l*a*b* colour space, 2008.

[5] International Electrotechnical Commission. Multimedia systems and equipment, colour measurement and management, part 2-1: Colour management, default rgb colour space srgb, 1999.

[6] Calvin Coopmans. Aggienav: A small, well integrated navigation sensor system for small unmanned aerial vehicles. In *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2009.

[7] Christian Dernehl, Dominik Franke, Hilal Diab, and Stefan Kowalewski. An architecture with integrated image processing for autonomous micro aerial vehicles. In *International Micro Air Vehicle Conference (IMAV)*, pages 138 – 145. IMAV, 2011.

[8] D. Eynard, P. Vasseur, C. Demonceaux, and V. Fremont. Uav altitude estimation by mixed stereoscopic vision. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 646–651, 2010.

[9] Felix Gathmann, Christian Dernehl, Dominik Franke, and Stefan Kowalewski. An integrated vision aided gps/ins navigation system for ultra-low-cost mavs. In *International Micro Air Vehicle Conference (IMAV)*, pages 1 – 8. IMAV, 2012.

[10] Farid Kendoul, Isabelle Fantoni, and Kenzo Nonami. Optic flow-based vision system for autonomous 3d localization and control of small aerial vehicles. *Robotics and Autonomous Systems*, 57(67):591 – 602, 2009.

[11] S. Lange, N. Sunderhauf, and P. Protzel. A vision based onboard approach for landing and position control of an autonomous multirotor uav in gps-denied environments. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–6, 2009.

[12] Swee King Phang, Jun Jie Ong, R.T.C. Yeo, B.M. Chen, and T.H. Lee. Autonomous mini-uav for indoor flight with embedded on-board vision processing as navigation system. In *Computational Technologies in Electrical and Electronics Engineering (SIBIRCON), 2010 IEEE Region 8 International Conference on*, pages 722–727, 2010.

[13] D. Salazar, M. Hernandez-Pajares, J.M. Juan, and J. Sanz. Rapid Open Source GPS software development for modern embedded systems: Using the GPSTk with the Gumstix. . Technical report, Grupo de Astronomia y Geomatica (gAGE), Universitat Politecnica de Catalunya, 2003.

[14] O. Shakernia, Yi Ma, T.J. Koo, J. Hespanha, and S.S. Sastry. Vision guided landing of an unmanned air vehicle. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, volume 4, pages 4143–4148 vol.4, 1999.

[15] J.L. Solka, D.J. Marchette, B. C. Wallet, V. L. Irwin, and G.W. Rogers. Identification of man-made regions in unmanned aerial vehicle imagery and videos. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):852–857, 1998.

[16] Fei Wang, Tao Wang, B.M. Chen, and T.H. Lee. An indoor unmanned coaxial rotorcraft system with vision positioning. In *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, pages 291–296, 2010.

[17] A.M. Waxman, J. Le Moigne, L.S. Davis, Eli Liang, and T. Siddalingaiah. A visual navigation system. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 1600–1606, 1986.

http://www.imavs.org/

Optic-flow based slope estimation for autonomous landing

G.C.H.E. de Croon, H.W. Ho, C. De Wagter,
E. van Kampen, B. Remes and Q.P. Chu

Micro Air Vehicle laboratory, Control and Operations department, Faculty of Aerospace Engineering, Delft University of Technology.
g.c.h.e.decroon@tudelft.nl

Abstract

In order to ease the lives of authors, editors, and trees, we present an easy-to-read guide to the easy-to-use easychair LaTeX2e document style class for EasyChair-based electronic and on-paper publishing of workshop and conference proceedings.

Abstract

Micro Air Vehicles need to have a robust landing capability, especially when they operate outside line-of-sight. Autonomous landing requires the identification of a relatively flat landing surface that does not have too large an inclination. In this article, a vision algorithm is introduced that fits a second-order approximation to the optic flow field underlying the optic flow vectors in images from a bottom camera. The flow field provides information on the ventral flow (v_x/h), the time-to-contact ($h/-v_z$), the flatness of the landing surface, and the surface slope. The algorithm is computationally efficient and since it regards the flow field as a whole, it is suitable for use during relatively fast maneuvers. The algorithm is subsequently tested on artificial image sequences, hand-held videos, and on the images made by a Parrot AR drone. In a preliminary robotic experiment, the AR drone uses the vision algorithm to determine when to land in a scenario where it flies off a stairs onto the flat floor.

http://www.imavs.org/

Contents

1 Introduction **1**

2 Vision algorithm for slope estimation **2**

3 Experiments on image sequences **5**

4 Landing experiment **8**

5 Conclusions **9**

1 Introduction

Autonomous landing is an important capability for Micro Air Vehicles (MAVs), especially if they have to operate outside line-of-sight. Active sensors such as a laser scanner [2] or multiple cameras as in stereo vision [1] would instantaneously provide distances to many points on the landing surface.

http://www.imavs.org/

However, such sensor setups only work at lower heights and are not energy or weight efficient. Since such efficiency is important for MAVs, it would be beneficial to have a robust landing strategy based on a single downward-pointing camera.

With a monocular camera setup, two main approaches to autonomous landing can be employed. The first approach is visual Simultaneous Localization And Mapping (SLAM) [6, 5], in which the 3D locations of all features in sight are determined. Various approaches to visual SLAM have been proposed over the years, which have consistently improved with respect to accuracy and computational effort [7, 19]. Still, SLAM delivers a lot of detailed information on the environment that is not strictly required for the landing task, and hence uses more computational resources than necessary.

The second approach is bio-inspired in the sense that it directly uses the optic flow field for control. The earliest studies of this approach [12, 16] are based on the biological finding that bees keep the ventral flow constant during a grazing landing [3, 4]. The ventral flow is equal to the translational velocity divided by the height (v_x/h), and keeping it constant results in a soft touch down. Since the ventral flow cannot account for the vertical dynamics, recent studies [13, 14, 17] complement the ventral flow with the time-to-contact, i.e., the height divided by vertical velocity ($h/-v_z$). The advantage of a bio-inspired approach is that light-weight neuromorphic sensors with high sensitivity and update frequency can be used directly for controlling the landing [9, 17].

The above-mentioned bio-inspired landing studies focus on autonomous landing on a flat surface. However, many real-world missions for MAVs will involve unknown landing sites that may be cluttered or have a slope. Indeed, bees can choose their landing site and adapt their body attitude to the slope of the surface on which they are landing [8]. For an MAV the detection of the landing surface's slope would also be of interest, either for adapting the MAV's attitude or for evaluating the suitability of the surface for landing.

In this article, a computationally efficient computer vision algorithm is proposed that uses a bottom camera. First, optic flow vectors are determined and then a second order fit of the entire optic flow field is made. The fit provides information on the ventral flow, time-to-contact, flatness of the landing surface, and surface slope. The algorithm is computationally efficient and robust to noisy optic flow vectors that are likely to occur in a real MAV landing scenario. In addition, it lends itself well for translation to algorithms for novel sensors that mimick insect eyes [18, 11].

The remainder of the article is structured as follows. In Section 2, the vision algorithm is explained. It is tested on image sequences in Section 3. The results of a landing experiment with a Parrot AR drone are discussed in Section 4. Finally, conclusions are drawn in Section 5.

2 Vision algorithm for slope estimation

The vision algorithm proposed for the slope estimation is based on the early optic flow work in [15]. In the explanation of the algorithm, a pinhole camera model is employed. The algorithm assumes (1) the camera to point downward, (2) the rotation rates to be known from the MAV's state estimation (e.g., using gyros), and (3) the landing surface in sight to be predominantly planar. Under these assumptions, the optic flow vectors in the image follow the equations:

$$u = (-U + xW)/Z, \tag{1}$$

$$v = (-V + yW)/Z, \tag{2}$$

with u and v the (derotated) optic flow in the x- and y-direction of the image, and U, V, W the motion in X, Y, and Z direction, respectively. x and y are image coordinates. The surface height Z can be modelled as a plane:

$$Z = R + \alpha X + \beta Y, \tag{3}$$

with R the height above the surface at $(x, y) = (0, 0)$. By a transformation of coordinates, in [15], the surface height is rewritten as:

$$z = (Z - R)/Z = \alpha x + \beta y, \tag{4}$$

and the velocities are scaled with respect to the height R, i.e., $u_0 = U/R$, $v_0 = V/R$, and $w_0 = W/R$. This leads to:

$$u = (-u_0 + xw_0)(1 - z) \tag{5}$$

$$v = (-v_0 + yw_0)(1 - z) \tag{6}$$

Replacing z with $\alpha x + \beta y$ (from Eq. 4), leads to:

$$u = -u_0 + (u_0\alpha + w_0)x + u_0\beta y - \alpha w_0 x^2 - \beta w_0 xy \tag{7}$$

$$v = -v_0 + v_0\alpha x + (v_0\beta + w_0)y - \beta w_0 y^2 - \alpha w_0 xy \tag{8}$$

In [15], the rotational optic flow terms were left in, and the discussion hence goes toward how to find the Focus-of-Expansion (FoE) and determine the first and second order spatial flow derivatives at that point. In that way, all terms can be identified (translational, rotational, and the slopes of the surface). However, determining the FoE is a difficult task in itself, and errors in its location can have a large influence on the subsequent results. In addition, since MAVs typically have access to gyro measurements, the rotational components do not have to be determined. So instead of finding the FoE, the vision algorithm proposed in this article immediately determines the parameters of the optic flow field:

$$u = \mathbf{pu}[1, x, y, x^2, xy]^T, \tag{9}$$

$$v = \mathbf{pv}[1, x, y, x^2, xy]^T, \tag{10}$$

The parameter vectors **pu** and **pv** are estimated separately with a maximal likelihood linear least squares estimate within a robust RANSAC estimation procedure [10], with 5 points per fit and 20 iterations. Figure 1 illustrates the result of such a fit.

The so-determined parameters can then be used to estimate the following variables of interest for an autonomous landing. The **ventral flow** is set to $(u_0, v_0) = (pu_1, pv_1)$. The **time-to-contact** and **slopes** can be retrieved from the first order spatial derivatives at the center of the image $(x, y) = (0, 0)$:

$$u_x = u_0\alpha + w_0 = pu_2 + 2pu_4 x + pu_5 y = pu_2, \tag{11}$$

$$u_y = u_0\beta = pu_3 + pu_5 x = pu_3, \tag{12}$$

$$v_x = v_0\alpha = pv_2 + pv_5 y = pv_2, \tag{13}$$

$$v_y = v_0\beta + w_0 = pv_3 + 2pv_4 y + pv_3 x = pv_3, \tag{14}$$

The slopes can then be retrieved as $\alpha = pv_2/v_0$ and $\beta = pu_3/u_0$. However, these equations become ill-conditioned and hence sensitive to even the smallest of noise if u_0 or v_0 are small. If there is insufficient motion in X and Y direction, then it may still be possible to estimate the slopes on the basis of the second order derivatives:

$$u_{xx} = -2\alpha w_0 = 2pu_4, \tag{15}$$

$$u_{xy} = -\beta w_0 = pu_5, \tag{16}$$

$$v_{xy} = -w_0\alpha = pv_5, \tag{17}$$

$$v_{yy} = -2\beta w_0 = 2pv_4, \tag{18}$$

http://www.imavs.org/

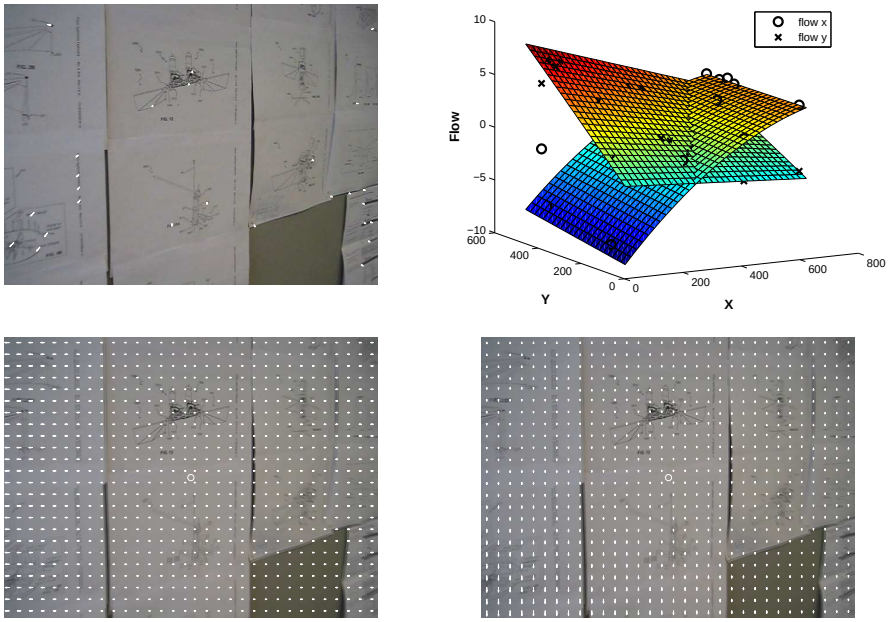


Figure 1: Illustration of the vision process. Top left: the determined optic flow vectors. Top right: quadratic fits for the flow in x-direction (circle markers) and y-direction (cross-markers). Bottom left: estimated flow field in x-direction. Bottom right: estimated flow field in y-direction. The motion of the camera is straight towards the wall, which has an angle of $\sim 45°$ to the camera axis.

which leads to two formulas for α and two formulas for β. However, all of these formulas depend on w_0, the relative vertical velocity. When looking at the formulas for the first order spatial derivatives, one will notice that w_0 cannot be determined without knowing α or β - seemingly introducing a chicken-and-egg problem. The solution lies in the realization that we only need these second order derivatives if the ventral flow estimate(s) are small. If, for example, $u_0 \approx 0$ then:

$$u_x = u_0\alpha + w_0 \approx w_0, \tag{19}$$

and, similarly, for $v_0 \approx 0$:

$$v_y = v_0\beta + w_0 \approx w_0, \tag{20}$$

In summary, when the horizontal flow is not sufficient, the relative velocity w_0 can be determined. In turn, this leads to slope estimates $\alpha = -pu_4/w_0$, $\alpha = -pv_5/w_0$, $\beta = -pu_5/w_0$, and $\beta = -pv_4/w_0$. All these equations become ill-conditioned if w_0 becomes small. This is intuitive, since if there is also little motion in the Z-direction, then no estimates of slope are possible. The remaining question is then how to deal with cases in which an MAV has velocities in multiple directions. For optimal estimation, one should fuse the different α and β estimates on the basis of their certainties and possible prior probability distributions. However, in this preliminary work slopes are determined on the basis of the first-order optic flow derivatives if there is sufficient motion in the X, Y-plane and only on the basis of the second-order optic flow derivatives if there is not.

http://www.imavs.org/

The time-to-contact τ is determined on the basis of the divergence $D = u_x + v_y$, with $\tau = 2/D$. Hence, τ includes the ventral flow and the slopes (see Eqq. 11 and 14). This makes sense, because in the case of pure horizontal motion, the MAV can still intersect the landing surface if it has a slope. However, if one is only interested in the component in the Z-direction, $\tau = 1/w_0$ could be used. In the current work the first definition is employed.

Finally, the **flatness** of the slope is related to how good the optic flow vectors fit with the above-described quadratic model. The RANSAC procedure returns a number of inliers and an error, which can both serve as measures for flatness.

3 Experiments on image sequences

In this section, a preliminary test of the vision algorithm is performed by applying it to various image sequences. The MATLAB code of the vision algorithm is available at `http://www.bene-guido.eu/`, so that the reader can test it on his / her own image sequences[1]. The algorithm is tested on three types of image sequences: (1) image zooms, (2) manually made videos, and (3) images from the Parrot AR drone. Figure 2 shows five example images.



Figure 2: Example images. From left to right: artificial image, images from manual videos (wall, flat scene, structured scene), image from the drone's bottom camera.

The test on image zooms serves to test the estimation of time-to-contact and ventral flow, and is motivated by the fact that in this setup the ground-truth values are known. Five image zooms were performed on a roof texture. The ground-truth time-to-contact decreases at a constant rate from 200 to 5. The results of the time-to-contact estimation are shown in Figure 3. The estimated time-to-contact is very accurate, even up to 200 frames from contact.

In order to test the slope estimation, three sets of videos have been made of a textured wall. In the first set the camera moves in the Y-direction (first up and then down), in the second set in the X-direction (first left and then right), and in the third set the camera moves in the Z-direction (in this case toward the wall). Put in a landing context, the first two sets cover horizontal motion, while the third set covers vertical motion toward the landing surface. Each set contains 5 image sequences in which the angle of the wall roughly iterates over the angles $\{0, 15, 30, 45, 60\}$ degrees. Please remark that the camera is moved in-hand, so additional motions and even rotations are present in the images. Since there is no clear ground-truth, this experiment mainly serves the goal of verifying that the slope α increases with an increasing angle to the wall. Figure 2 shows an example image from the sequence at ~ 45 degrees.

Figure 4 shows the results of this experiment for motion in the Y-direction (left plot) and X-direction (right plot). The plots show the estimated slope α (y-axis) over time (x-axis). The brightness of the lines are determined by the angle with the wall, with the darkest color corresponding to 0 degrees and the brightest color corresponding to 60 degrees. The lines are somewhat noisy, especially when the

[1]Please note that the code will be made available only after publication of the article.

http://www.imavs.org/

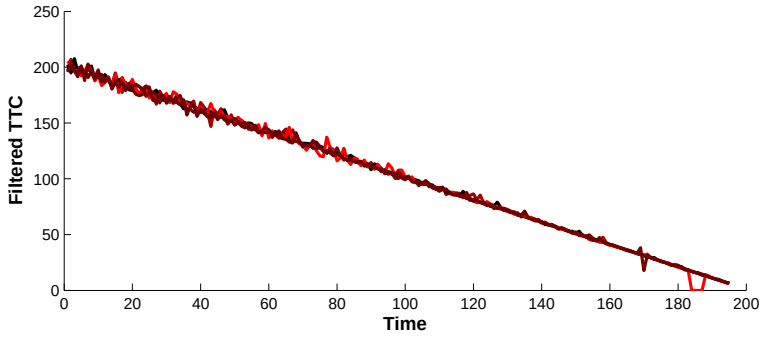


Figure 3: Results of time-to-contact estimation on five image zooms. The ground-truth time-to-contact goes from 200 to 5.

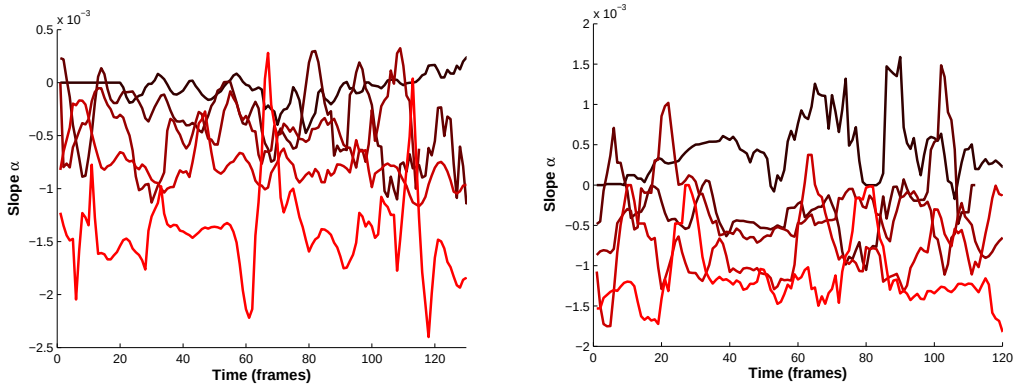


Figure 4: Results of slope estimation on video sequences. Left: motion in Y-direction. Right: motion in X-direction. The plots show the estimated slope α (y-axis) over time (x-axis). The brightness of the lines are determined by the angle with the wall, with the darkest color corresponding to 0 degrees and the brightest color corresponding to 60 degrees.

motion gets small, e.g., at reversal points of the movements. However, in both plots, a clear ordering can be seen from dark to bright slopes, as desired.

Figure 5 contains the results for motion in the Z-direction, illustrated in the same manner. In this case, the results are slightly less clear. The lower degree slopes switch between positive and negative values, while the higher degree slopes are constantly negative. The slightly less good results may be explained by the fact that the second-order spatial derivatives of the optic flow field are more difficult to determine than the first order ones.

Finally, the detection of flatness is tested with the help of two sets of each three videos. The first set contains flat surfaces, while the second set contains considerable 3D structure. The videos are made by moving toward the ground surface, i.e., in the Z-direction. The results are shown in Figure 6, which shows the mean absolute errors of the quadratic fits. The black lines illustrate the results for the flat surfaces, while the red lines illustrate the results for the scenes with 3D structure. The videos of flat surfaces have clearly a lower error than the videos of structured scenes. However,

http://www.imavs.org/

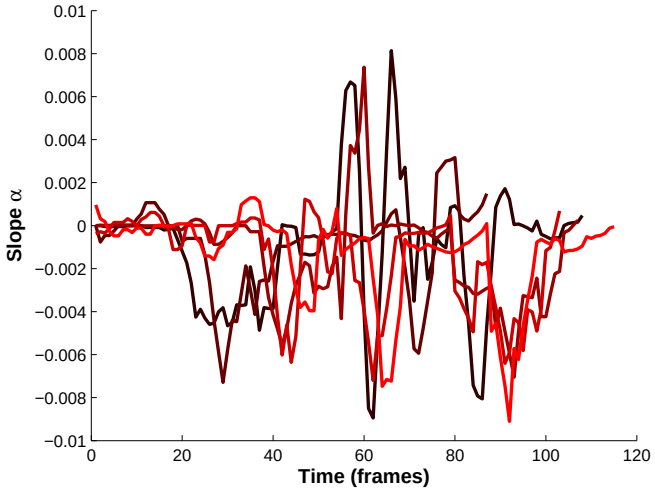


Figure 5: Results of slope estimation on video sequences with the camera moving in Z-direction. The plots show the estimated slope α (y-axis) over time (x-axis). The brightness of the lines are determined by the angle with the wall, with the darkest color corresponding to 0 degrees and the brightest color corresponding to 60 degrees.

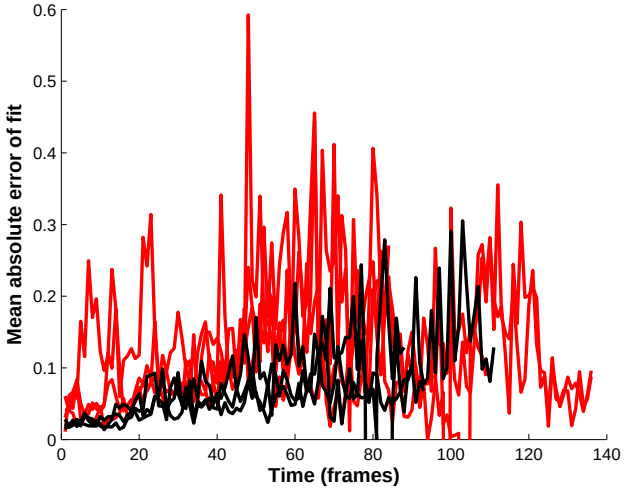


Figure 6: Results for estimation of landing surface flatness. Mean absolute error of the quadratic fits (y-axis) over time (x-axis). The black lines illustrate the results for the flat surfaces, while the red lines illustrate the results for the scenes with 3D structure.

placing a threshold is not straightforward, as the error also seems to depend on the time-to-contact. In particular, lower time-to-contacts entail larger optic flow, which leads to larger errors.

Finally, the slope estimation algorithm is run on the 160×120 images of the Parrot AR drone's

http://www.imavs.org/

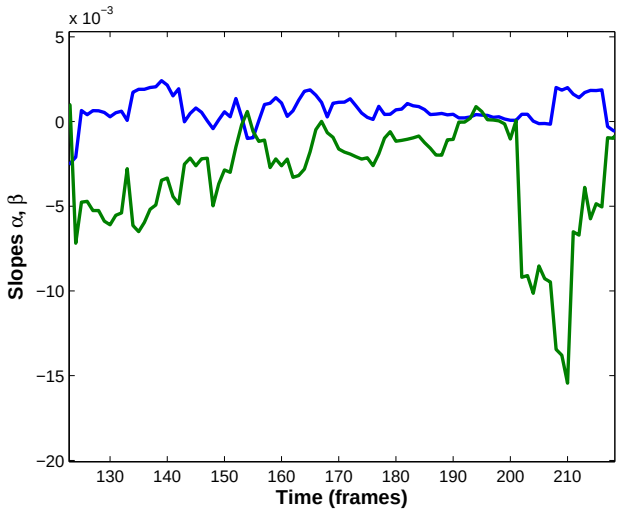


Figure 7: Results for estimation of slope with the Parrot AR drone's bottom camera. The plot shows α (blue) and β (green) over time (x-axis).

bottom camera, while it moves down a stairs. Figure 7 shows the estimated slope in α in the x-direction (blue) and β in y-direction (green). The y-direction is in the direction of the stairs. Indeed, while α is reasonably small and switches from positive to negative and back, β is larger and mostly negative.

In summary, the vision algorithm provides slope estimates that vary clearly with the angle of the wall and provides a cue for the detection of 3D structure below the MAV. The information is easier to obtain when the MAV makes movements in the X, Y-plane, as this then depends only on the first order spatial derivatives of the optic flow.

4 Landing experiment

The above-described landing algorithm is implemented in TU Delft's *SmartUAV* ground control station, and tested on a Parrot AR drone. In order to show that the landing algorithm can be used in a generic way, a staircase was chosen as the test environment. Stairs are challenging, since it is not a flat inclined surface. The setup for the experiment is presented in Figure 8. The experiment started with a forward motion of the drone (in the Y-direction) generated by a small pitch angle. This forward motion was controlled by proportional controller using the feedback of the velocity estimated from the AR drone optic flow measurement. The purpose of moving the drone in forward direction was to generate ventral flow to estimate the slope in real-time. The drone thus travels down the stairway and to a flat ground in the end. The height of the drone was under control of the Parrot firmware, implying that it remained at a fixed height above the steps of the stairs.

A simple landing strategy was used in this preliminary experiment, i.e. when the value of the estimated slope was close to zero $\beta \in [-0.0005, 0.0005]$, a landing command was given to land the drone immediately. The results of the estimated slope β from the onboard images captured during the flight is shown in Figure 9. It is clearly seen that the value of the slope was always negative when

http://www.imavs.org/

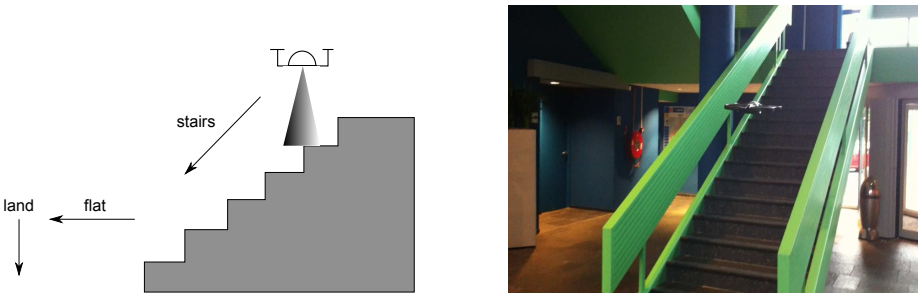


Figure 8: Landing Experiment Setup

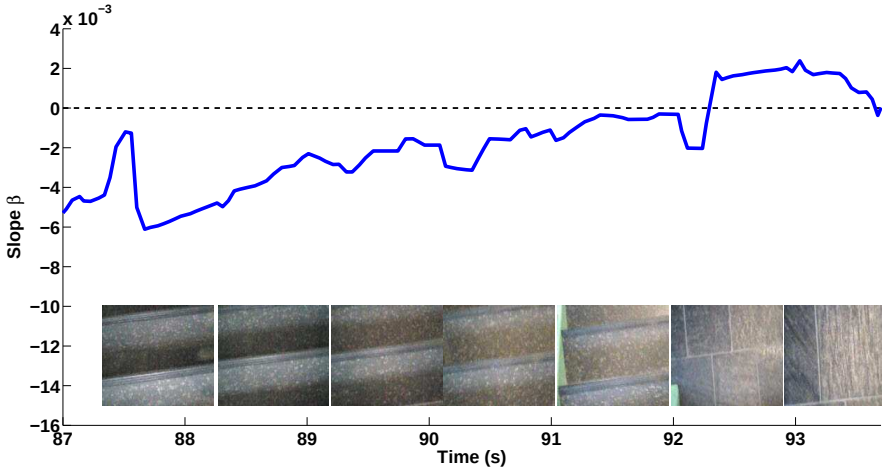


Figure 9: Real-time Slope Estimation Results with the Parrot AR drone's bottom camera. The plot shows the estimated slope β and images taken from the onboard camera over time (x-axis)

the drone was flying above the stairs. At the instant when it reached the flat surface, the value of the slope changed its sign and the autoland was activated to bring the MAV to the ground. Please note that the drone moves forward by pitching forward. The experimental results show that this considerably influences the estimated slope, which was not accounted for in the code. This led to slightly less negative slope values above the stairs and slightly more positive slope values above the flat floor[2].

5 Conclusions

The main conclusion is that the proposed vision algorithm is able to extract ventral flow, time-to-contact, flatness, and slope of the landing surface beneath an MAV. A preliminary landing experiment shows that the algorithm is computationally efficient enough to run in real-time and can discriminate between the stairs and a flat surface. The experiments show that it is important to take into account the pitch angle of the MAV while determining the slope of the surface.

[2]A video of the experiment can be found here: `http://www.youtube.com/watch?v=TXIPb1NvUJY&feature=youtu.be`

http://www.imavs.org/

Future work will focus on a further investigation of how optic flow can be used to best estimate the variables of interest. For example, combining slope estimates in a Bayesian fashion could considerably improve the estimates and allow for the exploitation of translation in multiple directions. Moreover, a more elaborate landing procedure utilizing the estimated variables will be devised and tested in a more complex environment.

References

[1] M.W. Achtelik, A.G. Bachrach, R. He, S. Prentice, and N. Roy. Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments. In *Unmanned Systems Technology XI, SPIE*, 2009.

[2] A. Bachrach, R. He, and N. Roy. Autonomous flight in unstructured and unknown indoor environments. In *EMAV, the Netherlands*, 2009.

[3] E. Baird, M.V. Srinivasan, S. Zhang, and A. Cowling. Visual control of flight speed in honeybees. *Journal of Experimental Biology*, 208:3895–3905, 2005.

[4] E. Baird, M.V. Srinivasan, S. Zhang, R. Lamont, and A. Cowling. Visual control of flight speed and height in honeybee. In *Simulation of Adaptive Behavior*, pages 40–51, 2006.

[5] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based MAV navigation in unknown and unstructured environments. In *IEEE International Conference on Robotics and Automation*, 2010.

[6] K. Celik, S.J. Chung, and A.K. Somani. Mvcslam: Mono-vision corner slam for autonomous micro-helicopters in gps denied environments. In *Proceedings of GNC'08*, 2008.

[7] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[8] C. Evangelista, P. Kraft, M. Dacke, J. Reinhard, and M.V. Srinivasan. The moment before touchdown: Landing manoeuvres of the honeybee apis mellifera. *Journal of Experimental Biology*, 213:262–270, 2010.

[9] F. Expert, S. Viollet, and F. Ruffier. Outdoor field performances of insect-based visual motion sensors. *Journal of Field Robotics*, 28(4):529–541, 2011.

[10] 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*, 24:381–395, 1981.

[11] D. Floreano, R. Pericet-Camara, S. Viollet, F. Ruffier, A. Brueckner, R. Leitel, W. Buss, M. Menouni, F. Expert, R. Juston, M.K. Dobrzynski, G. LEplattenier, H.A. Mallot, and N. Franceschini. Miniature curved artificial compound eyes. *Proceedings of the National Academy of Sciences*, 2013.

[12] N. Franceschini, S. Viollet, F. Ruffier, and J. Serres. Neuromimetic robots inspired by insect vision. *Advances in Science and Technology*, 58:127–136, 2008.

[13] B. Hérissé, T. Hamel, R. Mahony, and F.X. Russotto. The landing problem of a vtol unmanned aerial vehicle on a moving platform using optical flow. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1600–1605. IEEE, 2010.

[14] D. Izzo and G.C.H.E. de Croon. Landing with time-to-contact and ventral optic flow estimates. *Journal of Guidance, Control, and Dynamics*, 35(4):1362–1367, 2012.

[15] H.C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of Royal Society, London B*, 208:385–397, 1980.

[16] F. Ruffier and N.H. Franceschini. Aerial robot piloted in steep relief by optic flow sensors. In *(IROS 2008)*, pages 1266–1273, 2008.

[17] G. Sabiron, P. Chavent, T. Raharijaona, P. Fabiani, and F. Ruffier.

[18] Y.M. Song, Y. Xie, V. Malyarchuk, J. Xiao, I. Jung, K.-J. Choi, Z. Liu, H. Park, C. Lu, R.-H. Kim, R. Li, K.B. Crozier, Y. Huang, and J.A. Rogers. Digital cameras with designs inspired by the arthropod eye. *Nature*, 497:95–99.

[19] S. Weiss, D. Scaramuzza, and R. Siegwart. Monocular-slambased navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011.

http://www.imavs.org/

MAV Belief Space Planning In 3D Environments With Visual Bearing Observations

Laurie N. Bose[1]*and Dr. Arthur G. Richards[1†]

University of Bristol, United Kingdom
lb7943@bristol.ac.uk

Abstract

This paper examines the problem of path planning for a MAV in complex 3D environments without the use of GPS. Instead the MAV must rely on using sensor measurement of its surroundings for localisation. The quality of localisation and uncertainty in the MAV's estimated state will be determined by the path it takes through the environment. In order to minimize the probability of failure this state uncertainty must be incorporated into the planning process determining the path of the MAV. We present a path planner that can be used to produce paths which attempt to minimize the MAV's state uncertainty when operating in complex 3D environments. Simulation results are presented for a MAV equipped with a limited field of view camera sensor in environments featuring fixed localisation beacons from which, bearing measurements can be obtained and used for localisation.

http://www.imavs.org/

1 Introduction

An autonomous vehicle operating in a GPS denied environment must rely on sensor measurements of its surroundings for localisation within the environment. The quality and availability of such measurement may vary greatly across the environment with some areas allowing for far more accurate localisation than others. The path the vehicle takes through the environment determines the measurements received by its sensors and hence its ability to maintain good localisation with low uncertainty in its estimated state. The vehicle is less likely to be able to accurately follow paths which involve high state uncertainty which results in a higher probability of experiencing a collision or other form of potential failure. It becomes necessary to account for this uncertainty in the vehicle's estimated state when conducting path planning in order to produce paths which do not have a high probability of such failures occurring.

Belief space path planners are designed to account for uncertainty in vehicle state when generating paths. They produce paths providing a trade off between reducing the time taken and reducing the vehicle's state uncertainty. One such planner, the Belief Road map [6], is based on a modified version of the Probabilistic Road map [5]. Simulated results demonstrated the planners ability to produce paths minimizing uncertainty by travelling within close proximity to beacons which the vehicle could use for localisation. The Rapidly-exploring Random Belief Trees algorithm presented in [7] iteratively constructs a graph in belief space. This graph is then used to determine a safe path for the vehicle. Simulated results showed the vehicle's path deviate in order to pass through state measurement areas reducing the state uncertainty and enabling safely passage through narrow areas further along the path. The Particle RRT algorithm [8] accounts for uncertainty by stochastic simulating each tree expansion multiple times with process noise, adding multiple new branches per expansion.

The belief space planner introduced in this paper is geared towards navigation for MAV class vehicles equipped with camera type sensors, operating in complex 3D environments featuring many obstacles. Fixed beacons within these environments are used for localisation. The vehicle is able to take relative bearing measurements to a beacon provided it is both within the limited field of view of the vehicle's camera and not occluded by any obstacle. This setup is used as a rough representation of a MAV using a visual SLAM system to navigate through an already mapped area (with the beacons representing visual landmarks within the SLAM systems map). One of the long term goals of this work is to produce a belief space planner that fully utilizes a visual SLAM system for autonomous navigation.

*PhD Student, Department of Aerospace Engineering
†Senior Lecturer, Department of Aerospace Engineering

The planner operates on a graph which is determined by the construction of an Octree partitioning of the environment. The Octree's nodes are used to determine the positions of vertices within the graph. This process as discussed in [10], [9], [12], [13] results in an efcient vertex layout for path planning with sparse vertices in large open areas and dense clusters of vertices close to obstacles. Graph edges are then added between pairs of vertices that are within line of sight (LOS) of one another and whose associated Octree nodes are in contact. This step involves the expensive test of determining if two points are in LOS of one another by checking if the line segment connecting the two intersects any obstacle. The computational cost of this step is reduced by using the partitioning Octree to determine a subset of obstacles with which a connecting line segment could potentially collide. The planning algorithm involves incrementally constructing potential paths through the graph from an initial starting vertex. The construction of a path involves conducting a simulation of the MAV attempting to follow that path. This simulated MAV uses a particle filter for localisation and state estimation (as discussed in [4], [3], [2]). This particle filter captures the growth of uncertainty in the MAV's estimated state as it performs actions along with uncertainty reduction due to sensor measurements. At the end of the simulation the path is evaluated based upon both its length and in terms of minimizing uncertainty in the MAV's state estimation. Potential paths are then chosen to be extended or rejected based on this evaluation.

Section 2 introduces the problem formulation and representation used for the MAV's environment. Section 3 describes the construction of the graph which the planner operates within. Section 4 describes the full planning algorithm. Section 5 then presents a set of simulated results where paths produced by the planner attempting to maintain low state uncertainty are compared to paths which attempt to minimize distance traveled.

http://www.imavs.org/

2 Problem Formulation

The path planning problem involves finding a path from an initial position x_{start} to a destination x_{goal} which provides some desired trade off between minimizing distance traveled and maintaining low state uncertainty. The environment in which path planning is conducted is represented by a set of n convex obstacles S (non convex obstacles can be represented by decomposing into multiple convex) and a set of m fixed localisation beacons $B = \{b_1, ...b_m\}$ where $b_i \in B$ is the position of the ith beacon. The set $S = \{s_1, ...s_n\}$ consists of the convex polygon meshes representing each obstacle. For $s_i \in S$ we denote the set of vertices of the mesh $s_i.X$, the set of mesh edges $s_i.E$ and the set of mesh faces by $s_i.F$.

The MAV is taken to be equipped with a forward facing, limited field of view sensor which can take bearing measurements to the beacons of B. A measurement to a beacon $b_i \in B$ can be made provided b_i is both within the sensors field of view and is not obstructed by any obstacle in S. It is assumed that the MAV has access to a complete map of the environment. It can thus use the bearing measurements obtained from the beacons of B for localisation within this map. The map also gives knowledge of the obstacles in S allowing the MAV to account for them when path planning.

A graph across the environment is constructed as described in Section 3 consisting of a set of vertex positions V (including x_{start} and x_{goal}) and a set of undirected edges E dictating how the vertices are connected to one another. Potential paths through this graph are then represented by path nodes. A path node p has an associated vertex $p.x \in V$, a parent path node $p.p \in P$, current path length $p.l$ and an associated set of particle states $p.S$. Recursively iterating back through the parents of a path node p and examining their associated vertices traces out the whole path represented by node p. The set of particle states $p.S$ represents the belief of the MAV's true state upon travelling p's path and reaching $p.x$. Using a set of particle states to represent a probability distribution or "belief" for a true state is a standard technique used in Monte Carlo (particle filtering) localisation [2], [3], [8]. Path nodes also have a weighting $p.w$ which is generally based off both the node's path length $p.l$ and the uncertainty of its state belief represented by $p.S$, these two factors can be weighted differently based on the desired type of path (for example minimum distance paths would have zero weighting on uncertainty).

The planning problem then consists of finding a path through the graph connecting the vertex x_{start} to that of x_{goal} and which provides the desired trade off between distance and state uncertainty. The state uncertainty of the MAV upon reaching x_{goal} is dependent upon the path it traversed and the bearing measurements it took along the way which it used for localisation. The planner used to find such paths by searching the constructed graph is described in section 4.

Algorithm 1 Splitting Criteria for an Octree node

```
If node depth < max node depth
    For All s_i ∈ S
        For All edges e ∈ s_i.E
            If node intersects e
              return true
            End If
        End For
    End For
End If
return false
```

http://www.imavs.org/

3 Graph Construction

This section describes the algorithm used to construct the graph on which the path planner operates.

3.1 Determining Graph Vertices

The environment is first partitioned via the construction of an Octree whose nodes are then used to determine the locations of graph vertices in V. The Octree is initialized so that its initial node encompasses the entire volume in which path planning is to be conducted. Each node is then recursively subdivided into eight equally sized octants provided the node furfills a specific splitting criteria given in Algorithm 1. This splitting criteria produces a partition as shown in Figure 1a which has high node density at obstacle edges required for pathing around them but which ignores flat surfaces. During construction each node of the Octree also stores which obstacles from S it intersects with in order to facilitate determining edges between the vertices of V in a later graph construction step.

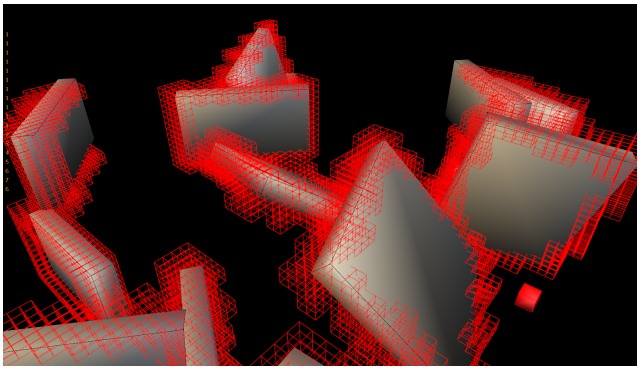
After construction of this Octree vertices are added to V at the centres of each of the Octree nodes (provided the node's centre is not inside any obstacle from S). Finally vertices are added at the initial start and goal positions $V = V \cup \{x_{start}, x_{goal}\}$. Note that the max node depth of the Octree in Algorithm 1 determines the level of detail to which the graph vertices V are placed throughout the environment. This affects the possible routes through the environment represented in the final graph. Important routes may not be represented if the max node depth is too low.

3.2 Determining Graph Edges

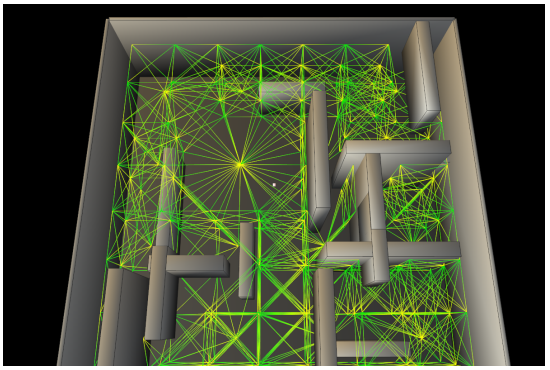
It is assumed that if any two graph vertices $x_A, x_B \in V$ are within direct line of sight then a safe path between them exists along the line connecting the two. This can be represented in the graph by the addition of an edge (x_A, x_B) to the set of edges E. However connecting every pair of vertices that are in line of sight leads to a huge amount of edges. Instead edges are added to E connecting pairs of vertices that are both within line of sight and whose associated Octree nodes are in contact with one another. This results in a graph of the form as shown in Figure 1b.

Determining if two positions x_A, x_B are within line of sight of each other requires checking that the line segment connecting the two does not intersect with any obstacle in S. A naive approach to performing this check is to simply check the line segment for intersection against each obstacle of S in turn until an intersection is detected or all obstacles have been checked against. However this method results in the number of intersection checks required simply growing with the number of obstacles present.

A preferable method involves using the constructed Octree to only perform intersection checks against a subset of the total set of obstacles. From the earlier graph construction in section 3.1, each Octree node stores a subset of obstacles with which it intersects. The method then consists of traversing the nodes of the Octree which the line segment passes through one by one and only performing intersection checks with obstacles that are contained within the subset of obstacles for



(a) The deepest nodes in a partitioning Octree showing how the splitting method biases the node density about obstacle edges



(b) Example of a graph constructed by the process described in Section 3. The partitioning Octree constructed with a low max node depth for clarity.

Figure 1

http://www.imavs.org/

one of these nodes. Traversing between the cells of the tree is generally a relatively cheap process especially in open areas which can be partitioned by a few large cells. This method generally results in far fewer intersection checks being performed and resulting in large performance gains.

3.3 Intersection Checks

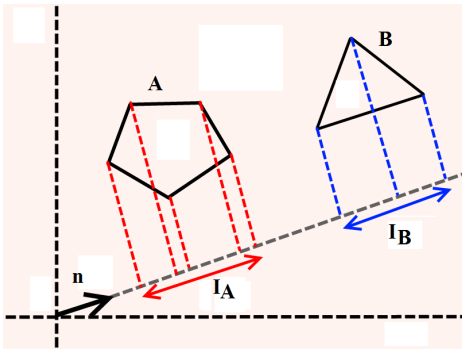
When constructing the Octree in Section 3.1 it is necessary to check for intersection between the cuboid nodes of the Octree and the convex obstacles of S. Intersection checks between convex objects, such as these are conducted with a method employing the separating axis theorem [11]. This simply states that if a plane can be placed between two convex objects such that each is fully contained on a different side of that plane then the objects are not intersecting. The normal to a such a separating plane is called a separating axis.

To determine if a specific direction $\hat{n}$ is a separating axis for two convex objects A and B, the projection of each of the convex objects onto $\hat{n}$ is calculated. This forms the projection intervals for each object I_A and I_B. If these intervals do not overlap then it can be seen that $\hat{n}$ is a separating axis for the object and hence they do not intersect. Figure 2 shows a 2D equivalent example of this.

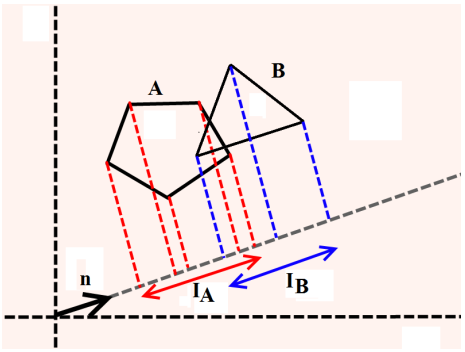
The interval formed from projecting a convex polygon mesh such as $s_i \in S$ onto a direction $\hat{n}$ is determined by calculating the dot product of $\hat{n}$ with each of the mesh's vertices $x \in s_i.X$. This then forms the set $D = \{x \bullet \hat{n} \mid x \in s_i.X\}$ and the interval formed by the projection is simply $I = [min(D), max(D)]$. If the convex mesh consists of a single edge (the object is a line segment) between two points p_1 and p_2 then its projection interval for a direction $\hat{n}$ is just $I = [min(D), max(D)]$ where $D = \{\hat{n} \bullet p_1, \hat{n} \bullet p_2\}$.

In order to fully determine if two convex meshes intersect, a number of directions must be checked to see if any provide a separating axis. This set of directions consists of the directions normal to the faces on each of the meshes along with all possible directions formed by taking the cross product between two edges, one from each mesh. Without checking the directions formed by the cross product of edges, separating axes such as those shown in Figure 2c would be overlooked. As soon as any of these directions are found to provide a separating axis the objects have been determined to not be intersecting and no further checks are necessary. On the other hand if no separating axis has been found after all these directions have been checked then the objects do intersect.

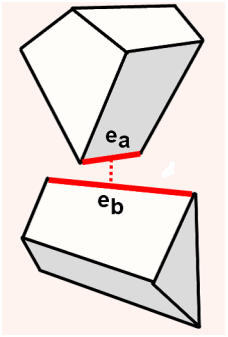
Note if a specific direction $\hat{n}$ has been checked to determine whether or not it is a separating axis it is unnecessary to check any other direction parallel to $\hat{n}$. For meshes such as those of the cuboid Octree nodes there will be faces whose normal direction is parallel to that of another face and edges parallel to others edges. It is important therefore to keep track of what directions have already been checked in order to avoid unnecessarily checking a direction parallel to one previous.



(a) Here the intervals I_A and I_B do not intersect indicating direction n is a separating axis for objects A and B and hence they do not intersect



(b) Here the intervals I_A and I_B intersect hence direction n is a not a separating axis however this alone is not enough to determine if A and B are truly intersecting



(c) Two objects with a separating axis shown by the dotted line whose direction is formed by the cross product of edges e_a and e_b

Figure 2: 2D Visualization of using vector projection to determine if a direction is a separating axis for two convex meshes

http://www.imavs.org/

4 Path Planner

Paths that feature high levels of uncertainty in the MAV's estimated state are generally undesirable as the MAV will be less likely to accurately follow such paths. This may then result in a collision or other form of failure due to the MAV straying from the desired path. The path planning method in this section attempts to find paths through the graph constructed in Section 3 which provide some desired trade off between maintaining low state uncertainty and minimizing distance traveled.

In order to evaluate a potential path the planner numerically simulates a MAV attempting to follow the path from vertex to vertex using a particle filter and bearing measurements to the beacons of B for localisation. The sample variance and covariance of the particle filter's set of particles is calculated upon reaching each vertex and used to determine the current uncertainty in the MAV's estimated state. This uncertainty along with the distance traveled so far, is then be used to evaluate the path upon reaching a vertex. If the path is deemed inferior to an already existing path to the current vertex then the simulation is halted and the path discarded.

4.1 Algorithm

The planner searches for paths through the visibility graph constructed in Section 3 consisting of the set of vertices V and edges E. A set of path nodes P (as described in section 2) is used to store different paths through the vertices of the graph. $U \subseteq P$ denotes the set of path nodes which should be used in attempting to create new paths by extending their current path to another vertex.

The set of path nodes P is initialized by setting $P = \{P_0\}$ where the path node P_0 is at the starting vertex ($P_0.v = x_{start} \in V$) and has a set of particle states $P_0.S$ representing the initial belief of the MAV's state. The parent of this node is simply initialized as $P_0.p = P_0$ as it is the path node from which all others originate. U is then initialized as $U = \{P_0\}$.

The planning algorithm used to generate paths for this setup is listed in Algorithm 2. The algorithm iteratively creates new paths through the graph from the path nodes in U. This involves selecting a path node $p \in U$ and attempting to extend p's existing path to every vertex $v \in V$ which is connected to $p.x$ by an edge from E (every vertex v such that $(v, p.x) \in E$).

For each path extension from $p.x$ to some other vertex $v \in V$ the MAV is simulated attempting to travel from $p.x$ to v taking $p.S$ as the initial belief of the MAV's state. The result of this simulation is then used to form a new path node q representing the extended path. This path node is also assigned a weighting $q.w$ based on both its uncertainty and path

Algorithm 2 Planning method

```
While U ≠ ∅
    p = minimum scoring element of U
    For all v ∈ V | (v, p.x) ∈ E
        q = Propagate(p, v)
        q.l = Assign_weighting(q)
        If !(∃a ∈ P | (q.w < a.w ∧ q.x = a.x))
            P = P ∪ p_test
            Insert(p_test)
        End If
    End For
    U = U \ p
End While
```

http://www.imavs.org/

length, each of which, are weighted by different constants (w_{len} and w_{un}) depending on the desired type of path. If this new path node is then deemed to provide the current best path to its vertex $q.x$ then it is added to both P and U.

After all possible extensions of a path node have been attempted, it is removed from U. Once U no longer contains any path nodes there are no more potential paths to investigate and the algorithm terminates.

There are a number of functions defined in Algorithm 2 which are now explained in greater detail.

4.1.1 $Propagate(p, v)$

The function $Propagate(p, v)$ takes a path node $p \in P$ and a graph vertex $v \in V$ and returns a new path node q such that $q.x = v$. The function carries out a numerical simulation of the MAV travelling from vertex $p.x$ to vertex v using a particle filter for localisation taking $p.S$ as the initial set of particles. At each step of this simulation the subset of beacons in B that are within line of sight of the MAV (if any) and which can be brought within the MAV's limited field of view sensor is determined. These beacons are then further examined to determine which would provide a bearing measurement resulting in the greatest decrease in state uncertainty, the MAV is then made to face this beacon. The set of particles on the returned path node $q.S$ is that of the particle filter at the end of the simulation.

4.1.2 $Assign\_weighting(p)$

This function assigns a weighting to a path node p based on both it's path length $p.l$ and an uncertainty measure u associated with its set of particle states $p.S$. In the set of results given for the planner, this uncertainty measure u was simply taken to be the sum of the x,y,z sample variances in the positions of the particle states in $p.S$. The path nodes weighting is then assigned as $p.w = w_{len} \times p.l + w_{un} \times u$ where w_{len} and w_{un} are constants that can be adjusted depending upon the desired type of path. For example setting $w_{un} = 0$ would assign weightings based purely on path length resulting in the planner attempting to produce minimum distance paths. On the other hand, setting w_{un} to a value much greater than w_{len} results in the planner producing paths which are far longer but which maintain much lower state uncertainty by taking measurements to the localisation beacons of B.

4.1.3 $Insert(p)$

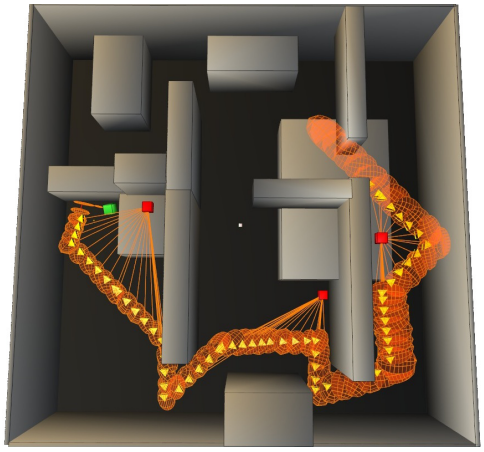
The function $Insert(p)$ inserts a path node $p \in P$ into the list of path nodes to update U. The position at which p is added to U is determined by its assigned weighting $p.w$ such that U maintains a list of path nodes ordered by their weightings. Low weighted path nodes in U can then be expanded first by simply choosing the last element of U for expansion. If we

were instead not to order U, or simply select which path node to expand at random, it would lead to an extremely large number of cases whereby a new path would be found that was slightly preferable to an existing path but which would then itself, be replaced by another slightly preferable path found soon after. This would result in a great amount of time wasted creating and examining similar paths which are likely to be replaced.

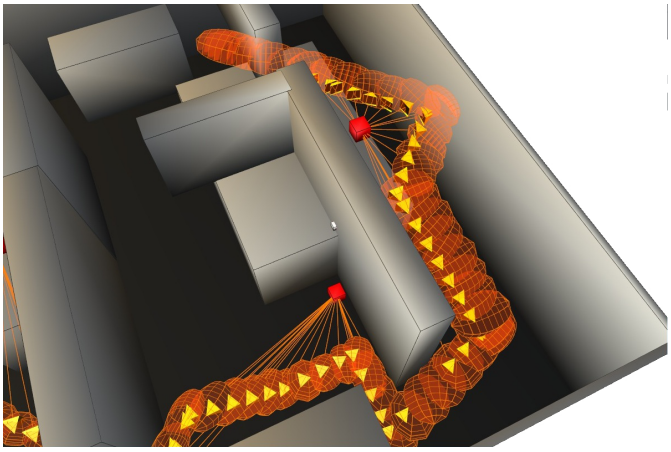
5 Results

A number of paths produced by the planner in different environments are now presented in this section (each environment is fully enclosed, however the roof of each is not drawn). The uncertainty of the MAV's estimated state is visualized by error ellipsoids formed from the set of particle states of the path nodes as discussed in Section 2. In each setup, robust paths produced by the planner when attempting to maintain low uncertainty (drawn in orange using weighting constants $w_{len} = 0.01$, $w_{un} = 1$) are compared with the paths produced when only minimizing distance (drawn in blue using weighting constants $w_{len} = 1$, $w_{un} = 0$). localisation beacons are drawn as red markers and measurements taken of them at points along a path are indicated by lines and vision cones.

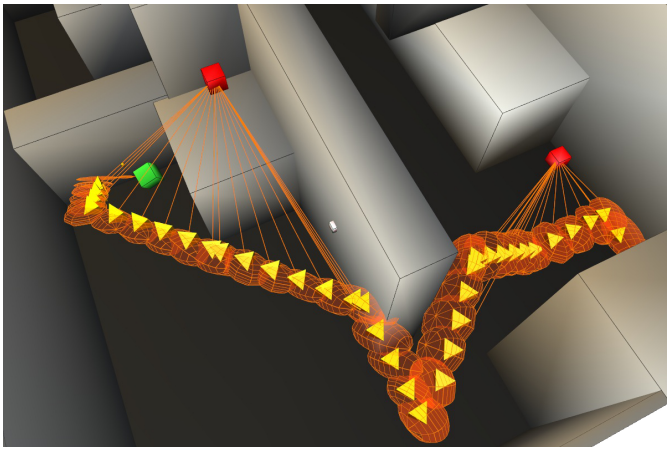
http://www.imavs.org/



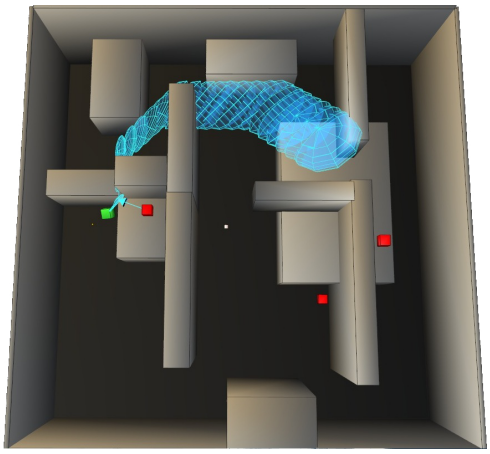
(a) Path produced by planning algorithm through complex environment



(b) Path of (a) shown from a different angle



(c) Path of (a) shown from another different angle

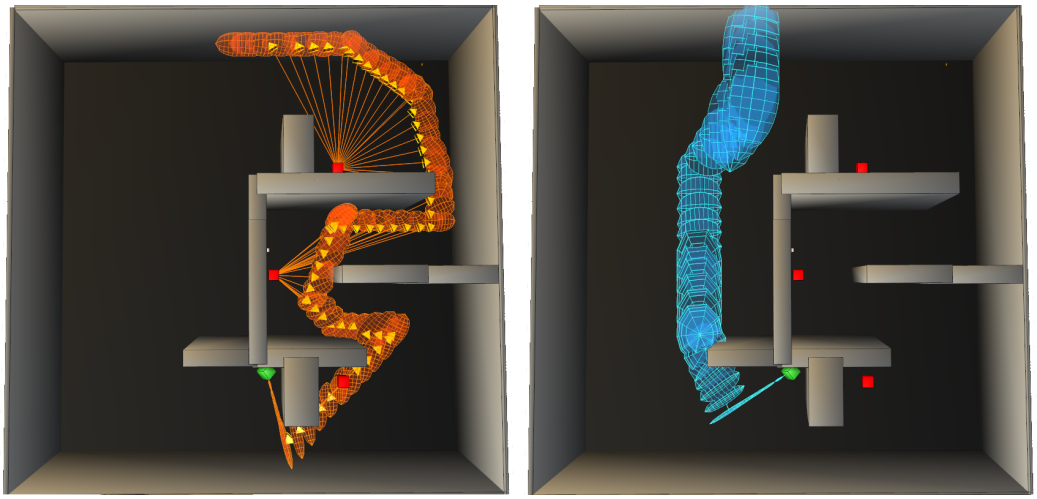


(d) Path Produced by planner when attempting to minimize path length

Figure 3

An example of a path generated by the planning algorithm between two points in a complex environment is shown Figure 3. The minimum uncertainty path is seen to take a route which enables it to obtain numerous beacon measurements, resulting in better localisation by the internal particle filter compared to the minimum distance path.

Examples of the planner operating in simpler environments are shown in Figures 4, 5, 6. Figures 4 and 5 show the planner producing paths which take the MAV through areas in which localisation beacons are always visible. Figure 5 in particular highlights how the minimum distance path risk collisions, as the ellipsoid cuts deep into the wall near the destination. Figure 6 shows the planner producing a path taking a very specific route through an open space which maintains line of sight between the MAV and a set of beacons in order to retain localisation. Tables 7b and 7c show the computation times and properties of the robust paths and shortest distance paths. The uncertainty measure of the robust path is seen to always be lower than that of the shortest distance path. This indicates that in every scenario the MAV is more likely to successfully follow the robust path over the shortest distance path.



(a) (b)

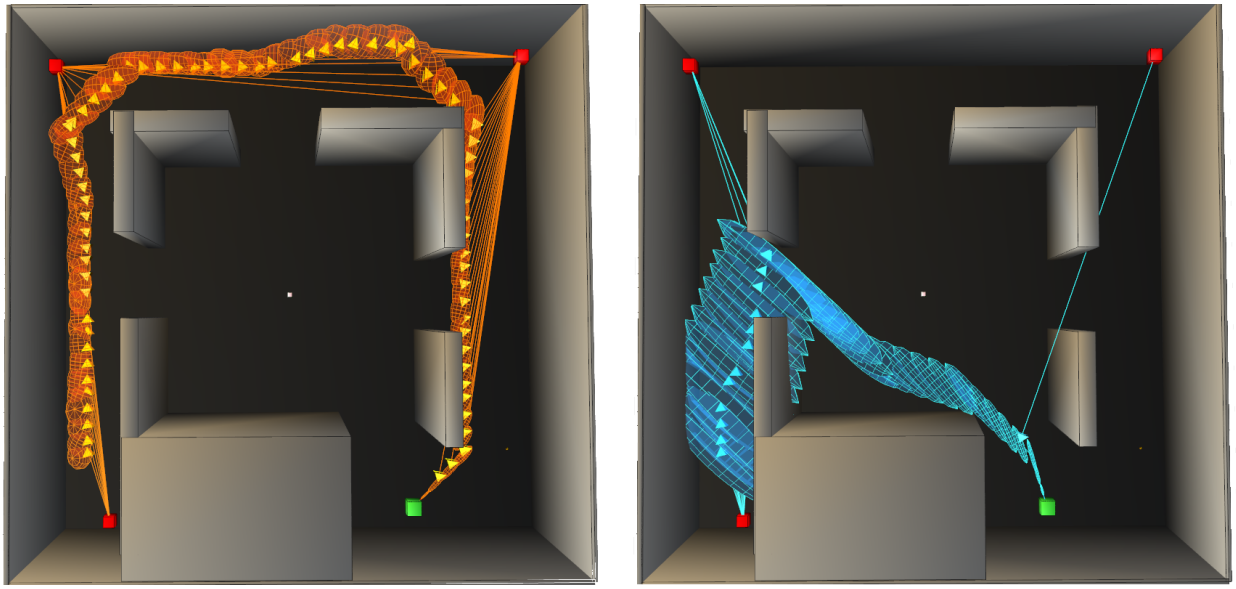
Figure 4: (a) Shows the path produced by the planner which attempts to maintain good localisation. By comparision (b) shows the path produced when the planner simply trys to minimize distance traveled. The path shown in (a) goes through the corridor to the right resulting in the MAV taking a longer route but able to observe several localisation beacons along the way.

6 Conclusions

This paper has addressed the problem of path planning for a MAV-like vehicle with the presence of state uncertainty. A belief space path planning algorithm is presented in which a particle filter is used to represent the belief of the MAV's state and examine how the belief would evolve along potential paths through the environment. The algorithm was demonstrated in simulation producing routes in complex 3-D environments which effectively maintained good localisation minimizing the probability of the MAV straying from the path.

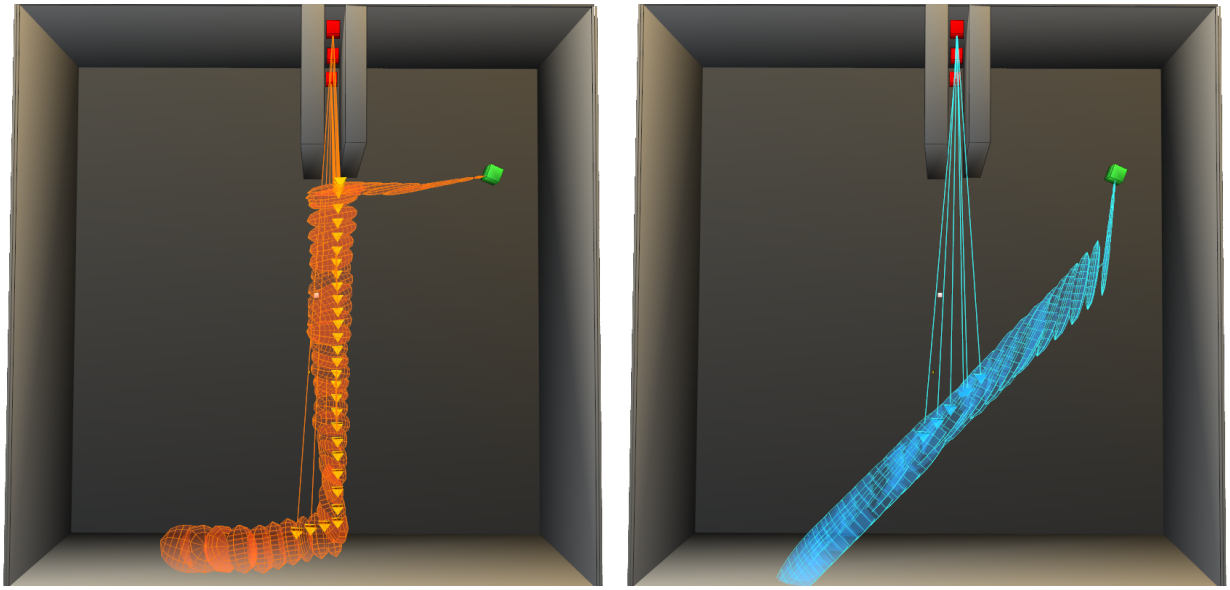
Much future work and examination remains, currently the planner does not explicitly take into account the probability of environmental collision when generating new paths, instead only performing comparisons to the current set of potential paths. Due to this omission many paths are generated which have a high likelyhood of collision. Though these paths are unlikely to be part of the final path outputted by the algorithm, significant computational time could be saved by culling such paths. By determining the probability of environmental collision at each step of path generation, the planner could also be formulated to produce paths that have a certain probability threshold of collision, rather than simply attempting to maintain a low state uncertainty.

http://www.imavs.org/



(a) (b)

Figure 5: Here (a) shows the planner producing a path which takes a long indirect route to the destination but which always keeps the MAV in line of sight of a measurement beacon. (b) Shows the planners route minimizing only distance.



(a) (b)

Figure 6: In this environment several localisation beacons are present placed at the end of a narrow corridor. The MAV is thus only able to observe them when it aligns itself with the corridor, bringing them within clear line of sight of its sensor. (a) Shows the planner producing a path diverging from the shortest route in order to align with this corridor and observe the beacons. (b) Shows the planners route minimizing only distance.

http://www.imavs.org/

Figure	Time (s)
3	6.14
4	2.88
5	5.41
6	1.18

(a) Algorithm computation times

Final Uncertainty Measure	Path length
18.81	687.36
32.18	655.89
21.26	707.39
46.73	418.11

(b) Robust path properties

Final Uncertainty Measure	Path length
125.41	320.74
173.92	324.50
267.35	355.87
63.82	270.64

(c) Shortest distance path properties

Figure 7

References

[1] Salomon, B. (2004). Fast Line-of-Sight Computations in Complex Environments.

[2] Fox, D., Thrun, S., Bur-, W., & Dellaert, F. (1998). Particle Filters for Mobile Robot localisation, 470–498.

[3] Thrun, S. (2002). Particle Filters in Robotics.

[4] Thrun, S., Fox, D., Burgard, W., & Dellaert, F. (2001). Robust Monte Carlo localisation for mobile robots. Artificial Intelligence, 128(1-2), 99–141. doi:10.1016/S0004-3702(01)00069-8

[5] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," IEEE Transactions on Robotics and Automation, vol. 12, no. 4, 1996.

[6] Prentice, S., & Roy, N. (2009). The Belief Roadmap: Efficient Planning in Belief Space by Factoring the Covariance. The International Journal of Robotics Research, 28(11-12), 1448–1465. doi:10.1177/0278364909341659

[7] Bry, A., & Roy, N. (2011). Rapidly-exploring Random Belief Trees for motion planning under uncertainty. 2011 IEEE International Conference on Robotics and Automation, 723–730. doi:10.1109/ICRA.2011.5980508

[8] Melchior, N. A., & Simmons, R. (2007). Particle RRT for Path Planning with Uncertainty, (April), 10–14.

[9] Hirt, J., Gauggel, D., Hensler, J., Blaich, M., & Bittel, O. (n.d.). Using Quadtrees for Realtime Pathfinding in Indoor Environments.

[10] Kambhampati, S., & Davis, L. (1986). Multiresolution path planning for mobile robots. IEEE Journal on Robotics and Automation, 2(3), 135–145. doi:10.1109/JRA.1986.1087051

[11] Gottschalk, Stefan. Separating axis theorem. Technical Report TR96-024, Department of Computer Science, UNC Chapel Hill, 1996.

[12] Kuwata, Y., & How, J. (2004). Three Dimensional Receding Horizon Control for UAVs,

[13] Herman, M. (n.d.). Fast, three-dimensional, collision-free motion planning. Proceedings. 1986 IEEE International Conference on Robotics and Automation, 3, 1056–1063. doi:10.1109/ROBOT.1986.1087622

http://www.imavs.org/

Scientific Session 3

Flapping wing MAVs

Chairman : Dieter Moorman

- Matej Karasek, Iulian Romanescu and Andre Preumont. ***Pitch Moment Generation and Measurement in a Robotic Hummingbird***
- João Caetano, Jochem Verboom, Coen de Visser, Guido de Croon, Bart Remes, Christophe de Wagter and Max Mulder. ***Near-Hover Flapping Wing MAV Aerodynamic Modelling - a linear model approach***
- Qi Wang, Hans Goosen and Fred Van Keulen. ***Optimal hovering kinematics with respect to various flapping wing shapes***

http://www.imavs.org/

Pitch Moment Generation and Measurement in a Robotic Hummingbird

Matěj Karásek[1], Iulian Romanescu[2] and André Preumont[1]

[1] Université Libre de Bruxelles, Brussels, Belgium
matej.karasek@ulb.ac.be, andre.preumont@ulb.ac.be
[2] "Gheorghe Asachi"Technical University Iaşi, Iaşi, Romania
iulian_romanescu@yahoo.com

Abstract

Micro Air Vehicles (MAVs) with flapping wings try to mimic their biological counterparts, insects and hummingbirds, as they can combine high agility manoeuvres with precision hovering flight. Near-hovering flapping flight is naturally unstable and needs to be stabilized actively. We present a novel mechanism for pitch moment generation in a robotic hummingbird that uses wing twist modulation via flexible wing root bars. A custom build force balance, sensitive enough to measure the cycle averaged pitch moment as well as lift force, is also presented. The introduced prototype mechanism generates pitch moment of up to ± 50g.mm. Finally we integrate a Shape Memory Alloy (SMA) wire to actuate the wing root bar ends. We present achievable displacement versus bandwidth as well as generated pitch moment.

http://www.imavs.org/

1 Introduction

Micro Air Vehicles (MAVs) or drones start to be part of our daily lives. Apart from obvious military applications they are being utilized more and more by police, fire brigades, film-makers as well as enthusiasts because they represent a relatively inexpensive yet powerful tool for aerial photo- and videography. The most widespread MAV design of today is a propeller-equipped quadrotor or multirotor, popular for its great stability and good manoeuvrability. Nevertheless, researchers keep exploring other possible ways of propulsion such as flapping wings inspired by insects and hummingbirds, the masters in combining precision hovering with breathtaking aerial acrobacy.

Many studies have shown that the near-hover flapping flight of diptera (flies with 1 pair of wings) is naturally unstable, e.g. [11, 4]. The animal brain needs to constantly process the sensory system signals and respond accordingly to maintain the desired body attitude. In other words, it needs to generate moments around the three body axis - roll, pitch and yaw.

This can be achieved through wing motion modifications that affect the lift and drag distribution over one flapping cycle [2]. Any asymmetry in the force distribution will cause non-zero cycle-averaged moments. If for example more lift is produced in front of the body than behind the body the average pitch moment over one wingbeat will be in nose-up direction. Roll moment can be achieved simply by a difference in left and right wing average lift production. Finally, an asymmetry between average drag forces of the left and right wing will result into yaw moment. Implementing similar mechanisms into flapping-wing robots, while keeping them relatively simple and, as a result, lightweight, is what makes their development challenging.

The first tail-less flapping wing MAV to demonstrate hovering flight was developed in 2012 by AeroVironment [5] (Figure 1a). It is controlled by a mechanism called wing twist modulation that modifies the wing camber. Recently three other flyers capable of hovering have been introduced. The

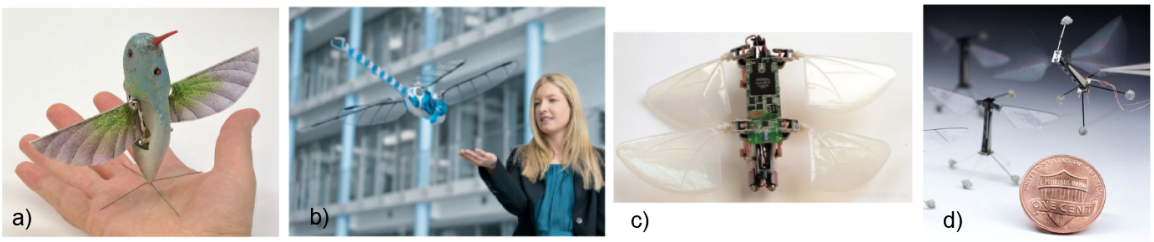


Figure 1: Existing flapping-wing robots capable of hovering: a) Nano Hummingbird (16.5 cm, 19 g), b) BionicOpter (63 cm, 175 g), c) TechJect Dragonfly (15 cm, 25 g), d) RoboFly (3 cm, 80 mg)

first two, BionicOpter by Festo [1] (Figure 1b) and Techject Dragonfly [7] (Figure 1c), use a four wing concept mimicking the dragonfly morphology. They are controlled by independent amplitude modulation of the 4 wings, moreover BionicOpter can also change their flapping plane orientation.

The last MAV is the piezoelectric-driven RoboFly from Harvard university (Figure 1d) that demonstrated the guided take-off already in 2008 [12]. The newest version [6] can be stabilized in air by controlling the amplitude and mean position of each wing independently, although the attitude sensing as well as power is still off-board.

In this work we present a novel mechanism for moment generation. It is using the wing twist modulation, similar to [5], however the mechanism is much simpler. We demonstrate that the prototype can generate lift force of 8.5g together with pitch moments of up to ± 50g.mm. Further, we explore the the possible use of Shape Memory Alloys (SMA) in control mechanism actuation.

Measurement of such small efforts in a highly vibratory system, which the flapping-wing robot certainly is, poses yet another challenge. To our knowledge none of the commercially available sensors combines sufficient sensitivity to both forces and moments with high natural frequencies preventing it from vibrating. For this reason we have also developed a 2DOF force-moment balance that is sensitive and yet stiff enough to evaluate the cycle averaged lift and pitch moment.

2 Robotic hummingbird prototype

The goal of our project is to develop a hummingbird sized robot with a wingspan of around 15 cm. The flapping motion is realized by a linkage mechanism that consists of two stages: a slider crank based mechanism, that generates a low amplitude rocker motion, and a four-bar linkage that amplifies the motion (Figure 2). The mechanism dimensions were optimized for the desired amplitude as well as for symmetry of upstroke and downstroke velocity profile. The gearbox has a reduction ratio of 19.75:1. The frame and the links are build by an Objet 3D printer (Eden series); the material used is DurusWhite. Aluminium and steel rivets are used to connect the links together. The mechanism is driven by a 7mm brushed DC motor.

The wings (Figure 3) are hand-build from a 15 micron thick polyester film; Carbon-fibre-reinforced polymer (CFRP) bars/bands are used as stiffeners. Their length is 70 mm and chord 25 mm. Sleeves at the leading edge and at the root edge (close to the body) are reinforced with Icarex to increase their durability. The sleeves allow easy assembly and disassembly as well as free rotation around the leading edge and root edge CFRP bars. Since the angle between the sleeves is greater than the angle between the leading edge and wing root bar the wing becomes cambered after assembly. The camber is, similarly to the Nano Hummingbird, twisted and bistable - it can passively flip from one side to

http://www.imavs.org/

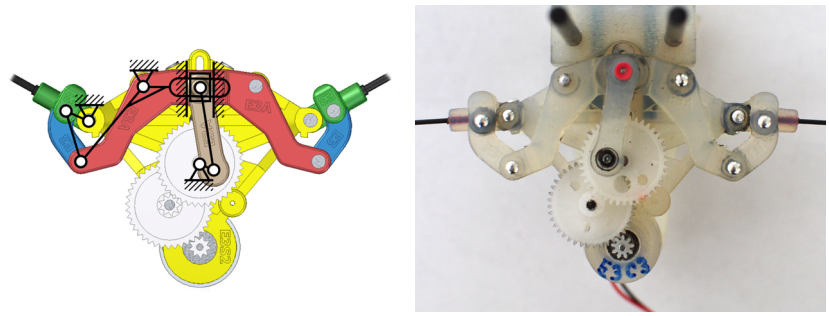


Figure 2: Flapping mechanism: model (left) and assembled prototype (right)

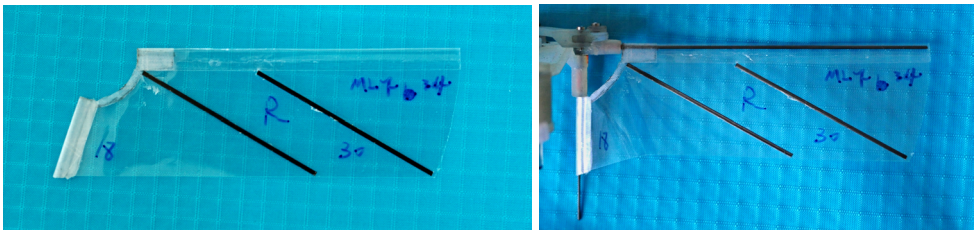


Figure 3: Polyester film wing becomes cambered after assembly.

another depending on the direction of motion.

Our best prototype so far can generate 12 g of lift while flapping at 25 Hz at a nominal voltage of 1 cell Li-Po battery (3.7 V). The robot weight, with a motor but without a battery, is approximately 8.5 g. A guided take-off with off board power has been successfully demonstrated (Figure 4).

2.1 Moment generation via wing twist modulation

The selected wing design allows to modify the wing twist and as a result the lift production by changing the angle between the leading edge and the root edge bar. This concept is used in the Nano

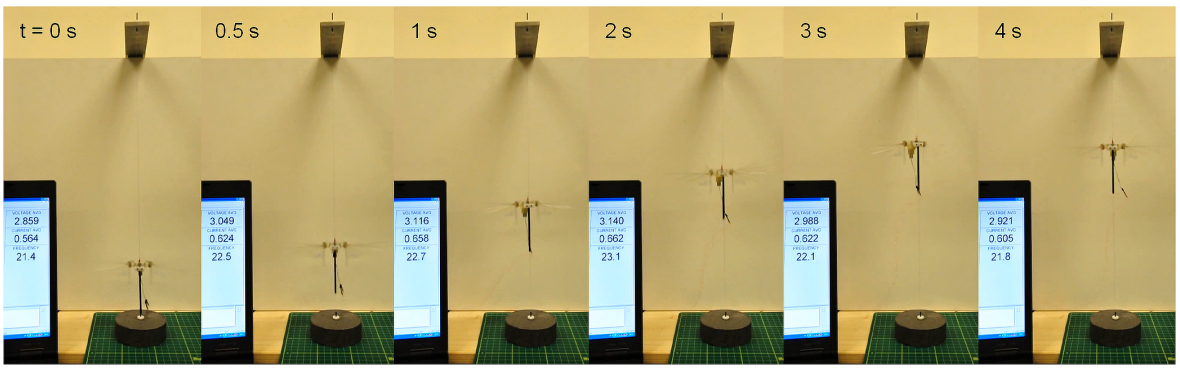


Figure 4: Take off demonstration with a guide-wire and off-board power

http://www.imavs.org/

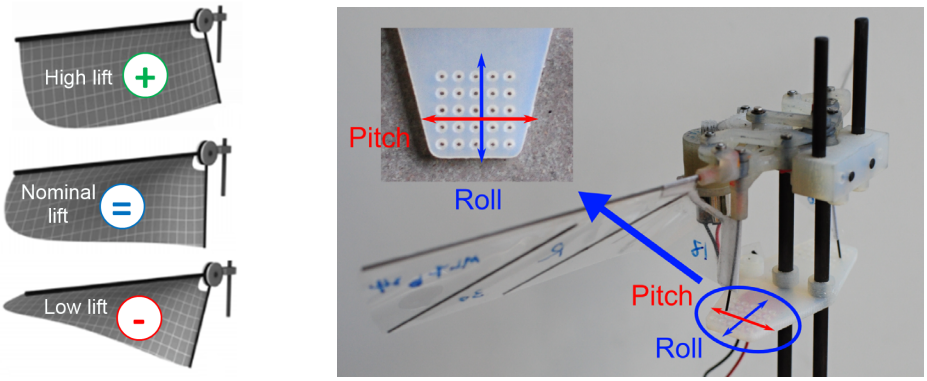


Figure 5: Wing twist modulation principle adapted from [5] (left), prototype for moment generation testing (right)

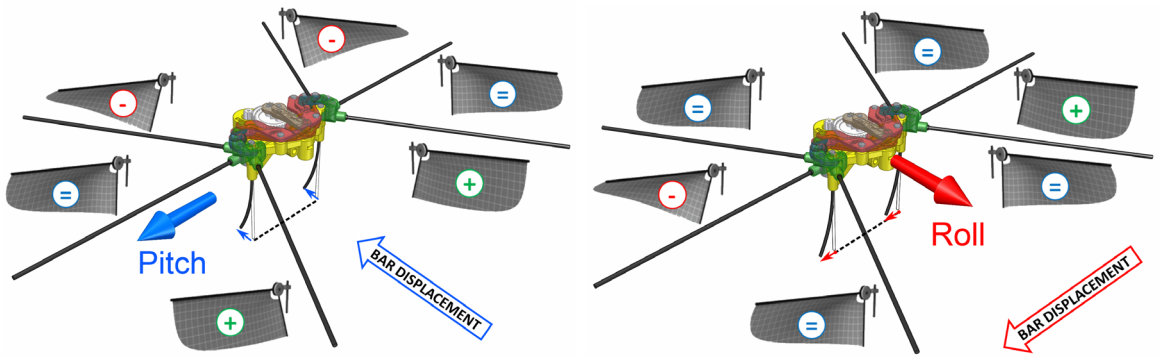


Figure 6: Moment generation via wing twist modulation: pitch moment due to front-back lift asymmetry (left), roll moment due to left-right lift asymmetry (right)

Hummingbird and is called the Variable Wing Twist Modulation [5]. For a wing with specifically optimized geometry, the lift force can be increased by moving the root bar away from the wing membrane and decreased by moving it towards the membrane (Figure 5 left).

To test this concept we have build a testing prototype (Figure 5 right). The CFRP root bars are clamped in the body frame but are flexible and their ends can be manually placed into a grid of equally spaced holes in the bottom part. The flexible bars simplify substantially the design as no joints are necessary.

The system works as follows: backward longitudinal displacement of the root bar end causes a twist (and lift) reduction when the wing is in front of the body but twist (and lift) increase when behind the body. If both left and right wing root bars are displaced in the same sense this results into a nose up pitch moment (Figure 6 left). If the root bars are displaced in the opposite sense a yaw moment is generated.

Lateral displacement of the root bars is used to generate roll moment (Figure 6 right). A deformation of one of the bars towards the body causes a decreased wing twist and thus lift reduction compared to the other wing.

http://www.imavs.org/

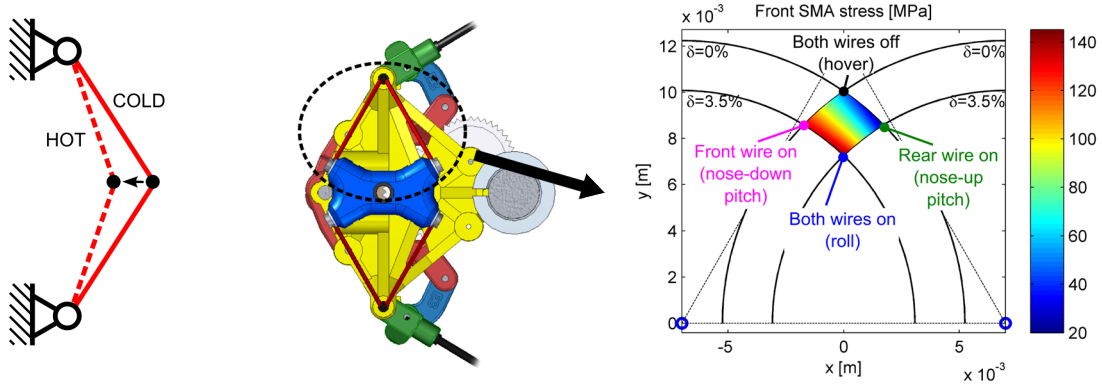


Figure 7: SMA driven control mechanism: kinematics for larger stroke (left), bottom view of the mechanism model (middle) with corresponding stress distribution for the front wire (right). The two supports are on top of each other in the bottom view.

2.2 SMA actuated control mechanism

The above solution is used to demonstrate and test the moment generation mechanism. In parallel we develop a control mechanism that can actively displace the bar ends to produce desired moments. The concept we present here is using Shape Memory Alloys (SMA) wires as actuators. The material uses a shape memory effect: when heated above certain temperature the crystal structure changes and, if the material is under stress, we observe contraction of the wire. After cooling the original shape is restored.

Advantages and disadvantages of using the SMA actuators in MAVs are discussed in [3]. We chose this actuator because it is very lightweight and provides directly a displacement. It can be heated simply by Joule effect. But it also has some limitations that need to be considered in the design: the maximal stroke is only about 5% of the wire length. The necessary (passive) cooling reduces significantly the bandwidth. Moreover, the material has a hysteretic behaviour due to phase transformation: the heating follows different characteristics than cooling. It can also suffer from fatigue, so operation at smaller strains (under 3.5%) and limited stresses (under 160MPa) is recommended [9].

The use of flexible bars instead of joints not only reduces the complexity, but the bar deformation also creates stress necessary for proper function of the SMAs. The small stroke achievable with a SMA wire can be overcome by the kinematics. We attach the SMA wire between two supports; the distance between these is just slightly shorter than the length of the SMA wire itself. Thus, a small contraction of the wire results in a relatively big displacement in the normal direction (Figure 7 left). The downside of this approach is that the maximal force is also reduced.

The designed system uses one pair of SMAs per wing that can displace the bar end in both longitudinal and lateral direction (Figure 7 middle). If only one of the wires is heated the bar moves diagonally in forward or backward direction, heating the two wires at the same time moves the bar laterally closer the body.

Heating the rear wires on both wings results into backward displacement of the bar ends and thus into a nose-up moment (as in Figure 6 left). Similarly heating the front wires on both wings results into a nose-down moment. Heating both wires on one wing while keeping them relaxed on the other wing results into a roll moment (similar to Figure 6 right).

The dimensions were selected to maximize the workspace while keeping the SMA wire stress under

http://www.imavs.org/

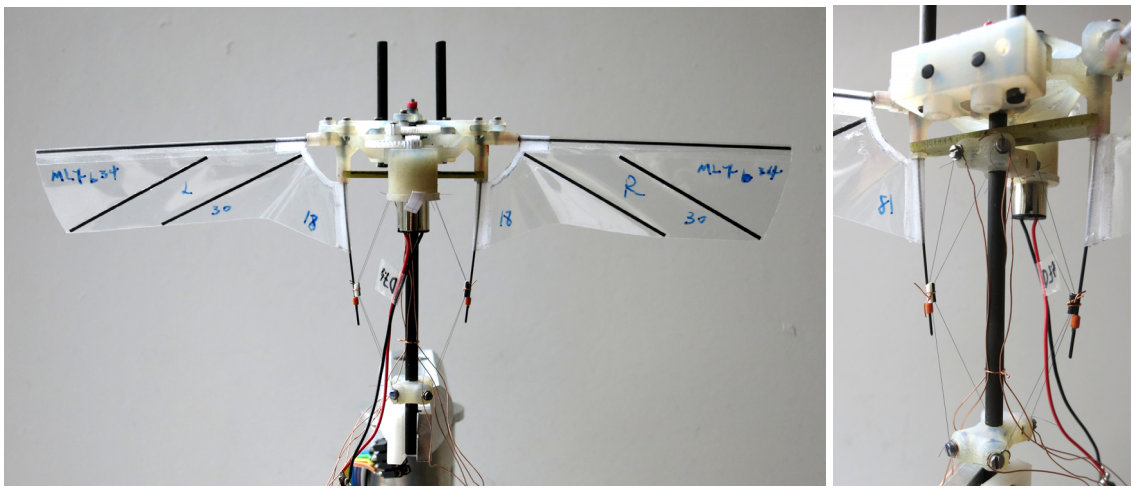


Figure 8: Robot prototype with SMA actuated control mechanism

the maximal recommended value yet high enough to assure proper phase transformation (Figure 7 right).

An important aspect that determines the mechanism bandwidth is the cycle time. While the heating phase can be accelerated by increasing the current, the cooling phase usually takes longer because the heat needs to be dissipated into the environment. The cooling is faster for wires with smaller diameter as the surface to volume ratio is higher. However, thinner wires mean also smaller maximal forces. The thinnest wire to withstand the estimated stress levels has a diameter of 50 microns.

The complete robot with the control mechanism is in Figure 8. The used SMA wires are SmartFlex®50μm [9]. Their active section is 53 mm long, the distance between the supports is 47 mm. The system to attach the SMA wires consists of two washers under the head of a bolt. The SMA wire goes around the bolt and is pressed between the washers. The power is brought by another cable, pressed by the second washer to the support.

3 Force balance

Measuring the efforts of a flapping wing robot is a challenging task. The generated forces are relatively small (order of 0.01 N) which requires high sensitivity. On top of that these efforts are of a periodic nature where not only the flapping frequency but also the higher harmonics are present. Hence the sensor should have a high resonance frequency.

The most frequently used commercial 6DOF force-torque sensor in the flapping wing research is the Nano-17 [10]. It is compact, it has a good sensitivity to forces (resolution of 1/1280 N $\approx$ 0.08 g) and high resonance frequency in all DOFs (7.2 kHz). However its sensitivity to moments (resolution of 1/256 Nm $\approx$ 390 g.mm) is much lower than what we need.

Since we did not find any other suitable commercial sensor we decided to design our own force balance. To keep the design simple we only want to measure the efforts in one plane (Fx, Fz, My). Moreover, we are primarily interested in the cycle averaged efforts.

In the past we already used a precision pocket scale to evaluate the mean lift with acceptable

http://www.imavs.org/

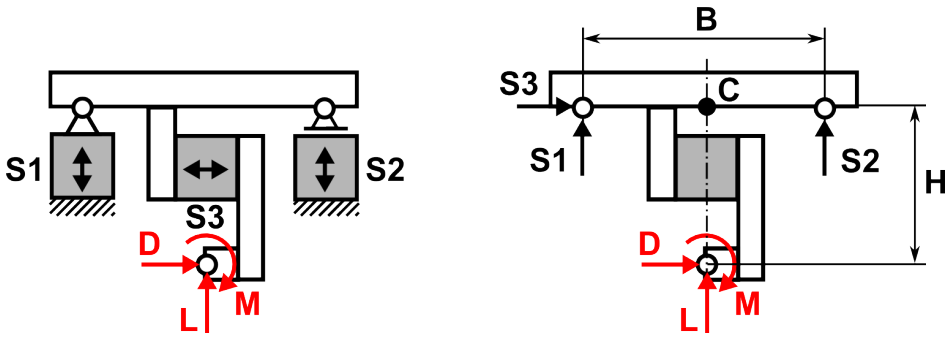


Figure 9: Schematics of the sensor (left) and its free body diagram (right)

results. The sensors used in the scales are usually double beam cantilevers with strain gages in full bridge configuration. Their advantage is that they are insensitive to the axial force as well as to the bending moment.

The experimentally determined resonance frequency of a sensor extracted from one of the scales was 210Hz, roughly 8 times the flapping frequency of our robot prototype. That is not enough to measure the time histories within one flapping cycle, but sufficient to evaluate the cycle averaged values. Thus we have selected these sensors as inexpensive yet reasonably precise base components for the designed force balance.

The balance uses three of these 1 axis force sensors in a configuration that is in Figure 9. Applying lift L, drag D and moment M on the balance results in the following sensor forces

$$\begin{aligned} S_1 &= (DH - M)/B - L/2 \\ S_2 &= (M - DH)/B - L/2 \\ S_3 &= D \end{aligned} \tag{1}$$

Sensor 3 measures directly the drag force, the sensitivity of sensors 1 and 2 to moment can be tuned by the selection of distance B and H. From the above equations we can express the measured efforts as

$$\begin{aligned} L &= -S_1 - S_2 \\ D &= S_3 \\ M &= S_3 H + (S_2 - S_1)B/2 \end{aligned} \tag{2}$$

During the preliminary tests we have noticed that the cycle averaged drag force was very low and its effect on the moment was negligible. Thus the third sensor has been dismounted for the measurements presented here. This allows to mount the robot closer to the rotation joints and increases the resonant frequency of the system. We measure the moment with respect to the centre between the two rotation joints, M_C, and the equation can be rewritten as

$$M_C = (S_2 - S_1)B/2 \tag{3}$$

The distance between the two sensor joints B was set to 50 mm, giving a good sensitivity yet enough space in between to fix the robot prototype. For small distance H and small cycle averaged drag force D the moment M_C is a good approximation of the true moment M, with an error that can be expressed from equation (2). This has no effect on the lift force.

http://www.imavs.org/

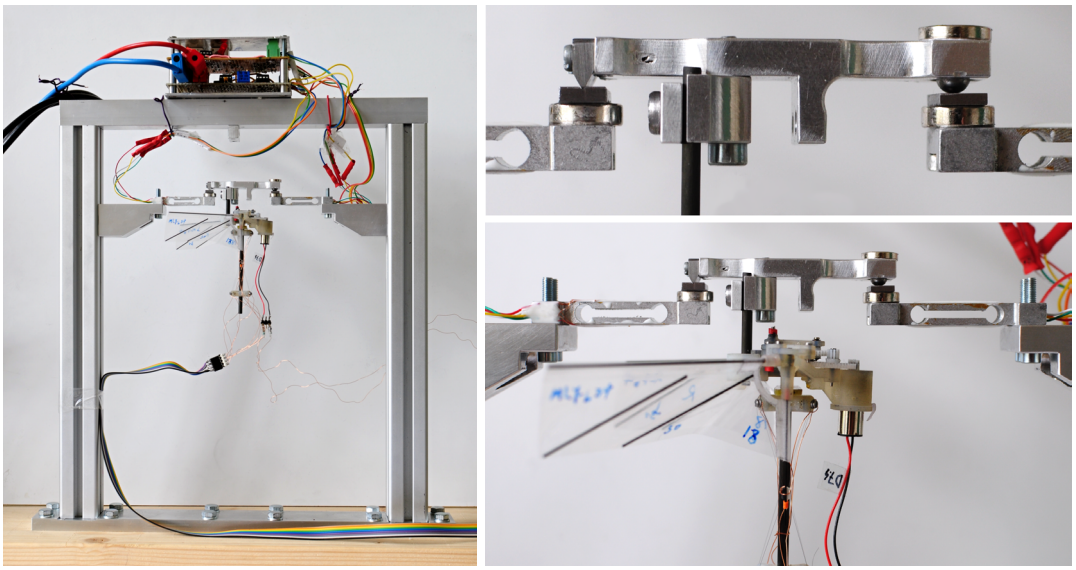


Figure 10: Force balance overview and detail views on sensors and magnetic joints

The assembled force balance is in Figure 10. Each sensor is connected to a custom build electronic circuit that provides stabilized power to the bridge and amplifies the bridge output. The sensors have been calibrated one at a time.

The rotational joints in the system should have as little friction as possible. The joint on the left is constructed as a blade inside a groove. The joint is held together by a magnetic force created by NdFeB cylindrical magnet that attracts the blade inside the groove. Both the blade and the groove are from soft magnetic steel. The joint on the right is build in a similar manner. Since it should also allow displacement to the sides, the blade was replaced by a steel ball that is touching a flat steel plate. Since the contact of the spherical ball and flat plate is only in one point, another magnet was attached on the top to increase the attracting force.

4 Experimental results

4.1 Force balance measurements

We process the force balance signals with a dSpace 1103 digital signal processor, together with the voltage and current readings of the DC motor. The flapping frequency can be detected from the motor current, because the motor torque is constantly changing due to the periodic aerodynamic and inertial forces.

The lift and moment are calculated using formulas (2) and (3). The system was designed only to measure the cycle averaged efforts due to its relatively low resonant frequency. The averaging is done online and is always calculated over a finite number of cycles. The averaging interval can be adjusted and was set to 5 seconds for the measurements presented here. The other readings (voltage, current, frequency) are averaged in the same way.

A typical lift measurement curve for a robot without control mechanism is in Figure 11. We performed the measurement 3 times in a row, each time we set the motor voltage to approximately

http://www.imavs.org/

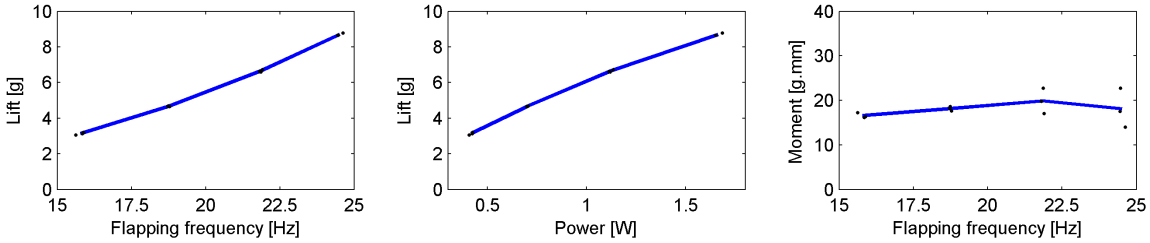


Figure 11: Force balance measurements: lift, necessary power and moment (measured values displayed as dots, lines are connecting average values).

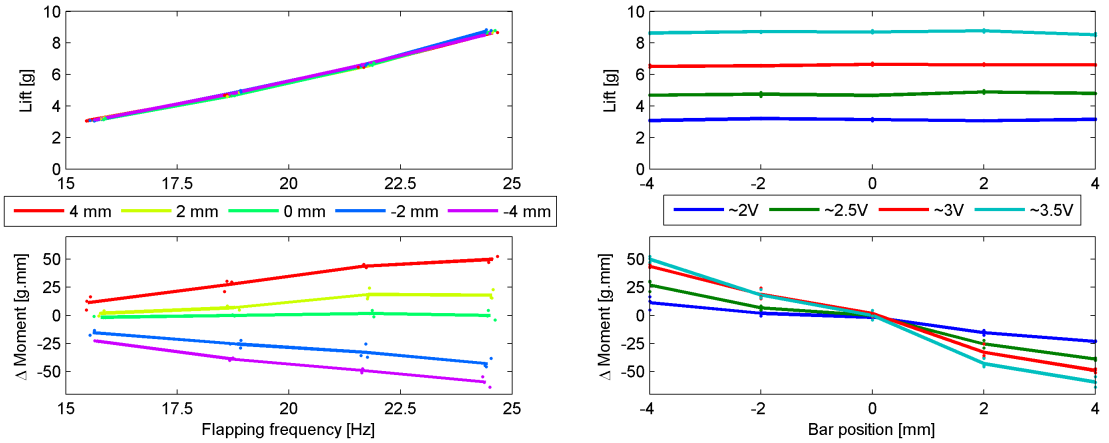


Figure 12: Lift and moment measurement results - passive bar displacement

2, 2.5, 3 and 3.5 V. The measured values are plotted as black dots, the blue line is connecting the average values. The repeatability of the lift measurements is good, with a typical standard deviation of below 0.1 g. We observe a bigger dispersion in the moment measurements, where a typical standard deviation is 5 g.mm.

4.2 Pitch moment generation

The main reason for designing the force balance was to demonstrate the pitch moment generation. The tests were performed with the control testing prototype with deformable wing root bars described in section 2.1 (Figure 5). The bar ends were fixed in 5 positions at 0, ±2 and ±4 mm from the centre where the root bars are straight. The measurements were carried out at 2, 2.5, 3 and 3.5 V and repeated 3 times for each position.

The results are plotted in Figure 12. Again, the individual measurements are plotted as dots and the average values connected by solid lines. The bar deformation has a negligible effect on the average lift force. The modulation of moment is approximately linear, but we can observe slightly different trend in positive and negative direction. We believe this is mostly caused by the asymmetric wing design where the stiffeners are glued only on one of the faces. When operating at 3.5 V, displacing

http://www.imavs.org/

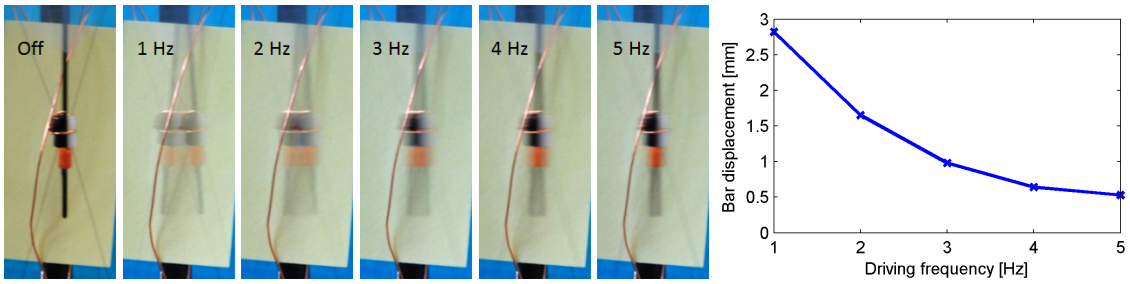


Figure 13: Long exposure images of root bar displacement by the SMA actuators driven at various frequencies (left) and processed results (right)

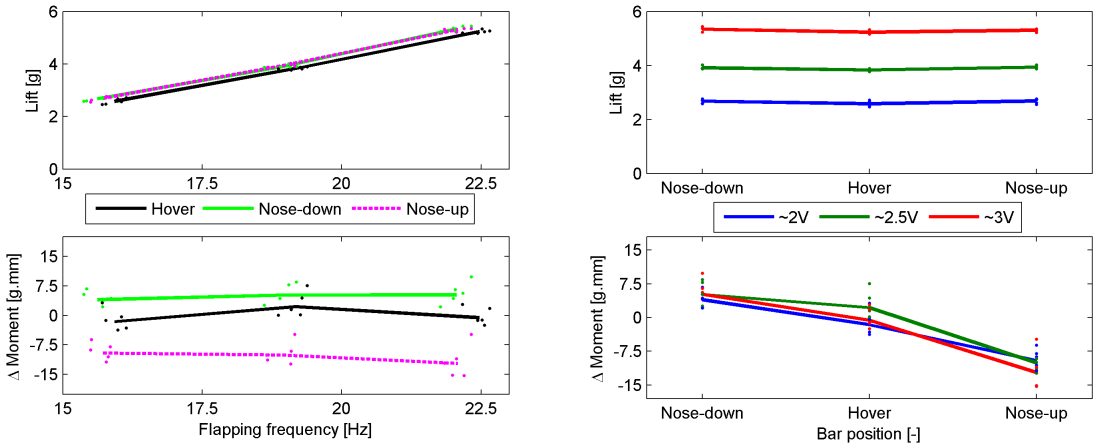


Figure 14: Lift and moment measurement results - SMA actuated bar displacement

the bars between -4 and 4 mm can generate a moment between -59 g.mm and 50 g.mm.

4.3 SMA driven control mechanism performance

The last tests were performed with the SMA actuated control mechanism. To determine the mechanism bandwidth we were periodically heating and cooling the front and rear pair of wires in an alternating manner, i.e. we were switching between the positions for nose-down and nose-up pitch moment. The achieved displacement was measured from a long exposure camera image. The duty cycle was 50% and the frequency was beeing changed from 1Hz to 5Hz. The current was constant during the heating phase, a value of 110mA was identified as optimal (no overheating). The airflow from the wings accelerated the cooling process.

The results presented in Figure 13 were measured at a moderate flapping frequency of about 16 Hz. The maximal displacement of 2.9 mm at 1 Hz decreases significantly as the command becomes faster. According to the results from the previous section the maximal displacement would generate a pitch moment of approximately ± 20 g.mm.

A direct pitch moment measurement was carried out in three (steady) positions according to

http://www.imavs.org/

Figure 7 right: 1) in hover position with all the wires relaxed, 2) in nose-up moment position with the rear pair of wires heated and 3) in nose-down moment position with the front pair of wires heated. In each position the measurement was repeated five times.

The results are plotted in Figure 14; individual measurements are displayed as dots and the lines represent the average values. The lift in hover position is slightly lower when compared to both nose-up and nose-down positions. This is in accordance with our expectations, because to generate a moment the wing root bar moves also in lateral direction which stretches the wing membrane.

The maximum generated moment is approximately -11 g.mm (nose-up) and 6 g.mm (nose-down), which is much lower than ± 20 g.mm estimated from the results in previous section. We assume this might be caused by smaller bar displacements at higher frequencies due to higher stress and higher cooling rates. However, direct comparison is not completely correct as the wing design in the SMA actuated prototype had to be modified to compensate for the wing root bar deformation, that needs to be present even in hover position to create pre-stress.

The asymmetry between nose-up and nose-down moments might come from imperfections of the hand build prototype (slight misalignments of the SMA supports, small variations of the SMA wires lengths, ...).

http://www.imavs.org/

5 Conclusions

We presented a new concept of pitch moment generation for a flapping wing MAV. It combines the wing twist modulation with flexible wing root bars. We demonstrated on a custom built force balance that it can generate a maximal cycle averaged pitch moment of ± 50 g.mm by displacing the bar ends by ± 4 mm.

To actively deform the wing root bars a SMA actuated control mechanism was developed. Currently it can displace the bars ends by almost ± 1.5 mm. The actuator can work at frequencies of up to 5Hz, however the displacement is significantly reduced. Direct measurements for static displacements show that pitch moments between -11 and 6 g.mm can be generated.

We have identified several weak points of the control mechanism that should be improved. Most importantly, the bandwidth as well as the achievable stroke (and thus moments) of the current system need to be increased, possibly by forced cooling and different kinematics respectively.

The attachment system of the SMA wires should be redesigned to simplify the adjustment process during the assembly and to improve the SMA lifetime, as many failures occurred there due to stress concentration. Currently we can only control the wires in an on-off manner which allows us to reach the "corners" of the workspace. A feedback control of the stroke might be implemented as the resistance of the SMA wires can be related to the strain.

As an alternative to the proposed solution, an actuator with larger stroke and higher bandwidth, such as a servomotor, might be used in future.

6 Acknowledgements

The research of Matěj Karásek is supported by FRIA grant from F.R.S. - FNRS in Belgium (FC 89554). The authors would also like to thank to our colleague Yanghai Nan and to internship students Neda Nourshamsi and Mathieu Dumas for their help with wing manufacturing and testing, sensor design and SMA mechanism testing.

References

[1] Festo AG & Co. KG. BionicOpter. http://www.festo.com/net/SupportPortal/Files/248133/Festo_BionicOpter_en.pdf, last viewed August 2013, 2013.

[2] Steven N. Fry, Rosalyn Sayaman, and Michael H. Dickinson. The aerodynamics of free-flight maneuvers in drosophila. *Science*, 300(5618):495–498, Apr. 2003.

[3] Giorgio Guglieri and Daniele Sartori. Experimental Characterization of Actuators for Micro Air Vehicles. *International Journal of Micro Air Vehicles*, 3(2):49–59, 2011.

[4] Matej Karasek and Andre Preumont. Flapping flight stability in hover: A comparison of various aerodynamic models. *International Journal of Micro Air Vehicles*, 4(3):203–226, 2012.

[5] Matthew Keennon, Karl Klingebiel, Henry Won, and Alexander Andriukov. Development of the nano hummingbird: A tailless flapping wing micro air vehicle. *AIAA paper 2012-0588*, pages 1–24, 2012.

[6] Kevin Y. Ma, Pakpong Chirarattananon, Sawyer B. Fuller, and Robert J. Wood. Controlled flight of a biologically inspired, insect-scale robot. *Science*, 340:603–607, 2013.

[7] Jayant Ratti, Emanuel Jones, and George Vachtsevanos. The hovering and gliding multi-wing flapping micro aerial vehicle. International patent application PCT/US2012/02571, Aug. 2012.

[8] Graham K. Taylor, Robert L. Nudds, and Adrian L. R. Thomas. Flying and swimming animals cruise at a Strouhal number tuned for high power efficiency. *Nature*, 425:707–711, Oct. 2003.

[9] SAES Getters. SmartFlex Wire & Springs. http://www.saesgetters.com/sites/default/files/SmartFlexWire&Springdatasheets_0.pdf, last viewed August 2013, 2009.

[10] SCHUNK GmbH & Co. KG. FT-Nano-17. http://www.schunk.com/schunk_files/attachments/FT-Nano_017_EN.pdf, last viewed August 2013, 2013.

[11] Mao Sun, Jikang Wang, and Yan Xiong. Dynamic flight stability of hovering insects. *Acta Mechanica Sinica*, 23(3):231–246, Jun. 2007.

[12] Robert J. Wood. The first takeoff of a biologically inspired at-scale robotic insect. *IEEE Transactions on Robotics*, 24(2):341–347, Apr. 2008.

http://www.imavs.org/

Near-Hover Flapping Wing MAV Aerodynamic Modelling - a linear model approach

J.V. Caetano[1,2] *, J. Verboom[1], C.C. de Visser[1], G.C.H.E. de Croon[1], B.D.W. Remes[1], C. de Wagter[1] and M. Mulder[1]

[1] Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands
[2] Portuguese Air Force Research Center, Air Force Academy, Sintra, Portugal

http://www.imavs.org/

Abstract

Flapping Wing Micro Air Vehicles typically have to be controlled actively in order to fly autonomously. Such active control benefits significantly from a model relating the flapper's state and control actions to the resulting forces and moments. This article presents the first results on system identification of a 17g ornithopter, the Delfly II. In particular, the Delfly II was programmed to perform specific flight manoeuvres in a high-fidelity sub-millimetre resolution external tracking system flight arena. The states were reconstructed and used to calculate the aerodynamic forces and moments that acted on the Delfly, under the assumptions of a rigid body and constant inertia properties, using general aircraft equations of motion. These forces and moments were used to identify parameters of a linear model around a trimmed hover condition. Two linear models are devised and compared: (1) a *full model* that incorporated state variables that were reconstructed from the tracking system, and (2) a *reduced model* that only included state variables that can be determined from onboard sensors. Using the linear time-invariant models it has been possible to estimate the aerodynamic forces with a very good approximation. However, neither the full nor the reduced models reveal to be able to estimate the aerodynamic moments as well as they do for the aerodynamic forces. The results point to the possibility of linear model use to control the Delfly in close-to-hover regime in closed loop.

1 Introduction

Flapping Wing Micro Air Vehicles (FWMAVs) pose many challenges, ranging from the understanding of their unsteady aerodynamics to designing the hardware. One of the greatest challenges is to achieve autonomous flight with ornithopters. Especially for nano FWMAVs, where flapping is believed to yield most advantages [1, 12, 14, 21], the small size and mass heavily restrict the possible sensors and processing onboard. At total take-off weights of only a few grams, attitude and flight control are still active areas of research [18, 26]. In this respect, a distinction has to be made between tailless and tailed designs.

Tailless designs offer the promise of performing highly dynamic manoeuvres as observed in small insects, such as bees and flies. However, these designs require active stabilization, as they are passively unstable. Although there is a considerable body of work on this type of FWMAV, only two such systems have shown actual flight ability: the Nano Hummingbird [18] and the Robobee [20]. The Nano Hummingbird utilizes onboard gyrometers and processing to perform onboard closed loop control on yaw, pitch and roll rates, positions of the yaw, pitch and roll servos, and the wing flapping frequency.

*Corresponding Author: j.v.caetano@tudelft.nl

Without human pilot input, the FWMAV performs autonomous attitude control on the basis of tuned control gains. The Robobee is extremely small and hence does not yet contain a power source, sensors, or processing onboard. It is able to hover and perform flight manoeuvres on the basis of a state estimate provided by external cameras. The control strategy for the attitude is based on a kinematic model and a corresponding Lyapunov function, resulting in a PD-like controller. The altitude and position are controlled with PD-controllers.

Tailed designs do not necessarily require active stabilization, because the tail acts as a natural damper. The centre of gravity can be placed such that close-to-hover flight becomes passively stable. Example studies using this set-up include [3, 8]. The advantage of this set-up is that research on such systems can focus on higher-level flight tasks such as altitude control [2, 10] or obstacle avoidance [9, 10]. A disadvantage of this set-up is that tailed FWMAVs are more sensitive to external perturbations. Moreover, it is very hard to maintain this stable configuration throughout the entire flight envelope without significant changes in, for instance, its centre of gravity. For this reason, active control is still necessary to enlarge the flight envelope, and to reject disturbances.

The previous approaches rely on manually tuned PID-controllers for the control of attitude and position. A model-based (nonlinear) control approach such as for example nonlinear dynamic inversion [24] could provide high performance control of the FWMAV over its entire flight envelope without the requirement for gain scheduling. The main reason behind the current absence of model-based control is the difficulty of designing a reliable model for FWMAVs. The principal difficulty derives from the unsteady aerodynamics [13] associated with flapping wing flight. Furthermore the flapping of the wings complicates the model description as their added inertia effects contribute to the dynamics [23] of the ornithopter. A complete modelling approach typically results [5, 23] in a complex nonlinear time-variant multi-body representation of the ornithopter. With the current state of technology it is not feasible to use such a model onboard for model-based control.

Aircraft system identification techniques have been applied before to a two wing ornithopter, with considerable success [17, 16]. However, this approach did not cover neither manoeuvre input design nor automated control over the inputs. More recently Caetano et al. have been able to program a FWMAV to perform automatic manoeuvres for system identification purposes [7] and calculate the aerodynamic forces and moments that act of the flapper, using flight path reconstruction and general aircraft equations of motion [6].

The **main contribution** of this article is centred around the development of computationally inexpensive linear model for the onboard control of the Delfly II FWMAV, which lead to the development of linear time-invariant aerodynamic models of the flapper that are valid for stable flight conditions. In particular, a high-fidelity sub-millimetre resolution external tracking system was used to directly measure the position and attitude of the Delfly II. The flight states were then reconstructed and used to calculate the aerodynamic forces and moments that acted on the flapper, under the assumptions of a rigid body and constant inertia properties (Section 2). Two linear models were devised and compared: (1) a *full model* that incorporates state variables reconstructed from the tracking system, and (2) a *reduced model* that only includes state variables that may be measured or calculated directly from the existing onboard sensors (Section 3). The quality of the results of these models if presented in Section 3.2. The final conclusions are drawn in Section 4.

2 System Overview

The present section describes the systems that were used for the flight tests and the respective data processing.

http://www.imavs.org/

2.1 Delfly II Micro Air Vehicle

The Delfly II [11] is a bio-inspired ornithopter, configured with 4 flapping wings and an inverted "T" tail. It weighs only 17g and is capable of performing hovered flight as well as transition to a maximum forward flight speed of 5m/s and vice-versa. The Delfly that was used in the flight tests was equipped with a Radio Control (RC) receiver for manual operation and a programmable autopilot that is capable of performing stable automatic flight at a trimmed configuration. A full description of the Delfly can be found in [7] and [8].

2.2 Experimental Setup

The flight tests were conducted at the United States Research Laboratory Micro Air Vehicles Integration and Application Institute flight test chamber, that is equipped with a tracking system capable of recording the position of small reflective markers at 200Hz. In order to track the Delfly's position, eight markers were placed on the flappers' structure, as presented in Figure 1. In order to capture the full dynamic response of the Delfly over several dynamic frequencies, a set of in-flight longitudinal (elevator) inputs was designed and pre-programmed into the autopilot. This way the inputs were performed thoroughly by the autopilot with, assuring its constant duration and flight regime, that ranged from -0.25m/s to 1m/s ground speed. The elevator inputs were designed as steps, doublets and triplets with a reference time of $\frac{1}{3}$ of a second. More considerations about the markers' positions and the manoeuvre input design are presented in [7] and [6].

3 System Identification Framework

This section presents the system identification framework that was used to go from the tracking system data to the linear models for the aerodynamic forces and moments.

3.1 Flight Path Reconstruction

The first step in the system identification framework is the reconstruction of the Delfly states from the recorded position data in time. This process is called flight path reconstruction [22] and aims to reconstruct states that cannot be measured directly, such as the angle of attack, or to refine the accuracy of sensor measured states.

3.1.1 Reference Frames

In order to obtain the Delfly's states, each of the markers' coordinates has to be described in the Delfly's body frame. For the sake of simplicity, the tracking system's reference frame will be addressed as being the inertial frame. Hence, the inertial (subscript I) frame's z_I axis is vertical and positive up, and x_I and y_I axes are horizontal and form a right-handed orthonormal frame with z_I. The body reference frame (subscript b) was defined using an intermediate "marker" reference frame, as the markers were placed on top of the body structure and not *in* the body structure. The $\overrightarrow{x_b}$ axis is positive forward; the $\overrightarrow{z_b}$ was defined by the cross product of the unit vector of $\overrightarrow{x_b}$ with the unit vector that is defined by the left-to-right wing marker vector ($\overrightarrow{W_{LR}}$), thus $\overrightarrow{z_b} = normalized(\overrightarrow{i_b} \times \overrightarrow{W_{LR}})$. The $\overrightarrow{y_b}$ axis was then defined by the cross product of $\overrightarrow{z_b} \times \overrightarrow{x_b}$. The origin of the body frame is at the Delfly's Centre of Gravity (CG). Figure 2 presents both the inertial ($\overrightarrow{x_I}, \overrightarrow{y_I}, \overrightarrow{z_I}$) and body frames ($\overrightarrow{x_b}, \overrightarrow{y_b}, \overrightarrow{z_b}$).

The rotation matrices between the frames can be devised by using a direct cosine matrix. Eq. 1 represents the rotation matrix from the inertial to the body frame, in the way that a vector can be

http://www.imavs.org/



Figure 1: Markers' positions on the Delfly.

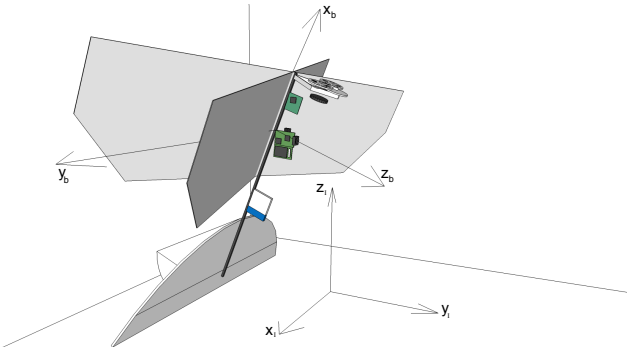


Figure 2: Inertial and body reference frames

http://www.imavs.org/

calculated in the body frame by multiplying the rotation matrix Rot_{bI} with that vector's coordinates written in the inertial frame.

The Euler angles can be determined for each time step, by corresponding the, commonly called, 321 rotation matrix [4] to the markers coordinates in the inertial frame. The 321 rotation follows the *yaw* (ψ) $\longrightarrow$ *pitch* (θ) $\longrightarrow$ *roll* (ϕ) sequence. One can easily obtain Rot_{bI} as a function of (ϕ, θ, ψ) by multiplying $Rot(\phi)Rot(\theta)Rot(\psi)$. The result is presented in Eq. 2. Hence, the (ψ, θ, ϕ) angles can be determined by relating the entries in Eqs. 1 and 2.

$$Rot_{bI} = \begin{bmatrix} i_{b_x} & i_{b_y} & i_{b_z} \\ j_{b_x} & j_{b_y} & j_{b_z} \\ k_{b_x} & k_{b_y} & k_{b_z} \end{bmatrix} \tag{1}$$

$$Rot_{bI} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & \sin\phi\cos\psi - \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix} \tag{2}$$

3.1.2 State Reconstruction

The Euler definition is limited to specific intervals [4] – for example, the θ angle is only defined in the interval $[-\frac{\pi}{2}; \frac{\pi}{2}]$. For the Delfly case, in a near-hover configuration, an elevator input can induce the transition to an inverted flight regime, where $\overrightarrow{z_b}$ is pointing upwards. In this situation, the ψ and ϕ have a sudden variation of 180°, as presented in Figure 5. In order to keep the Euler relations to be used in the general aircraft equations of motion (EOM) [15, 6], the Euler angles were converted to attitude angles ψ_y (yaw), θ_p (pitch) and ϕ_r (roll), as if these were measured on-board. This allows the dynamic equations to be defined over the ±90° pitch angle range and still remain in the same general form, not needing alternative formulations in terms of, e.g., quaternions. This way, angular rates (p, q, r) were obtained using Eq. 3. The linear velocities and accelerations, as well as the body angular accelerations ($\dot{p}$, $\dot{q}$, $\dot{r}$) were obtained by differentiating the markers positions and body rates, respectively, with respect to time. Table 1 presents all the states that were computed – they follow the commonly accepted definition in aeronautical engineering [15].

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta_p \\ 0 & cos\phi_r & sin\phi_r cos\theta_p \\ 0 & -sin\phi_r & cos\phi_y cos\theta_p \end{bmatrix} \begin{bmatrix} \dot{\phi}_y \\ \dot{\theta}_p \\ \dot{\psi}_r \end{bmatrix} \quad (3)$$

Euler Angles	Attitude Angles	Velocities	Accelerations	Angular Body Rates	Angular Body Accelerations	Aerodynamic Angles	Inputs
ϕ	ϕ_r roll	u	$\dot{u}$	p	$\dot{p}$	α	δ_f (flap.freq.)
θ	θ_p pitch	v	$\dot{v}$	q	$\dot{q}$	β	δ_e (elevator)
ψ	ψ_y yaw	w	$\dot{w}$	r	$\dot{r}$	——	δ_r (rudder)
——	——	V	——	——	——	——	——

Table 1: Reconstructed states, using the flight data and respective control surface inputs.

Some of the reconstructed states are presented here for a near-hover elevator doublet input manoeuvre. In order to decrease the influence of the high oscillatory modes induced by the flapping frequency (comprehended between 12Hz and 13Hz for the present test case), the states were filtered using a 3^{rd} order zero-phase lag Butterworth low-pass filter, with cut-off frequency of 10Hz - data processing also revealed that neglecting the information of the fast high order harmonics just above the flapping frequency improves for the forces and moments estimation. This technique is also applied onboard, when reading from the inertial measurement unit.

Figures 3 to 6 present a part of one of the flight tests, zoomed around the doublet input on the elevator, where it is possible to observe the somewhat constant states prior to the input, as well as the Delfly's behaviour until the oscillatory movement is fully dampened. The elvator input had a 2/3 of a second, with each commanded deflection lasting 1/3 of a second. The abscissa axes are defined in seconds. In these Figures the blue lines represent the full dynamic oscillatory behaviour of the states without any filtering; the red lines represent the filtered states; *and* the green line in Figure 5 represents the attitude angles, presented in Table 1.

It is possible to follow the manoeuvre by observing Figures 3 and 4. The input starts at second 5.85 of the flight test. The rudder oscillation (Figure 3) during the longitudinal manoeuvre is related to the tail's rotation and bending, as no input over the rudder was performed by the autopilot. Figure 6presents the angle of attack (top) and side slip angle (bottom). The velocity component u (in Figure 4) decreases to a negative value as the Delfly loses lift and height. The oscillation in w is due to the pitch manoeuvre – when w goes to negative means the Delfly is flying in the opposite direction, in inverted flight. As expected, v does not suffer considerable variations during the longitudinal manoeuvre. The velocity variation between seconds 6 and 7 in Figure 4 corresponds to the inverted flight, where the Delfly is flying "belly up". The inverted flight can be better seen in Figure 5, by following the pitch angle on the middle graph (depicted in green), while it surpasses the 90˚ value.

3.2 Aerodynamic Model Identification

The second step in the system identification framework is aerodynamic model identification. The aim of aerodynamic model identification is to create a model that relates control actions to resulting aerodynamic forces and moments. The resulting aerodynamic models can then be used in model-based flight control systems.

3.2.1 Initial Assumptions

Creating accurate *full flight envelope* aerodynamic models of aircraft is a highly challenging task, especially when the aircraft is subject to significant nonlinear aerodynamics. However, if we focus on

http://www.imavs.org/

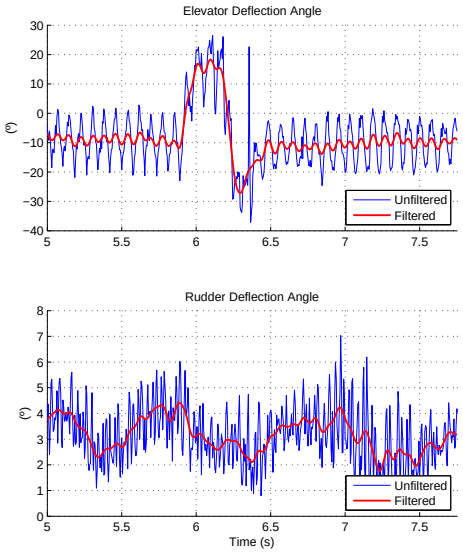


Figure 3: Elevator and rudder deflection angles, defined as positive up and right, respectively.

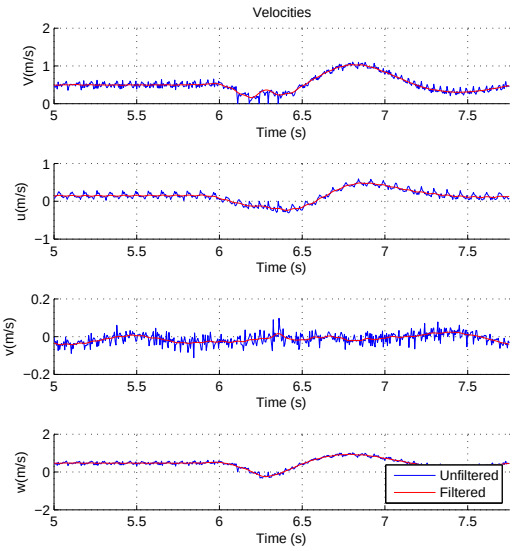


Figure 4: Body velocities in m/s. V corresponds to the free stream airspeed, decomposed in u, v and w for each of the body axes ($\overrightarrow{i_b}, \overrightarrow{j_b}, \overrightarrow{k_b}$).

small excitations around a stationary (trim) position, the nonlinear aerodynamics can be approximated with linear models [19]. In this work, we assume that Delfly is flying in a stationary condition and that the excitations around this condition are small. Under these assumptions, the linear modelling approach is valid.
Additionally, the dynamics of the Delfly are assumed to be described by the Newton-Euler equations of motion found in [19] or [6]. The assumptions involved are (1) rigid body kinematics; (2) no flapping (the flapping is then modelled as a thrust force); (3) constant mass and no inertia changes due to flapping or bending; (4) flat Earth; (5) no wind.

3.2.2 Model Structure Selection and Parameter Estimation

Several linear model structures of the form $X = X_{0_s} + \sum_{s=1}^{n} X_s.S$ were devised using a linear states' relation. Here the left-hand side term X represents the forces and moments obtained from the Newton-Euler equations of motion [4, 6]; the first term on the right-hand side, X_{0_s}, is the affine coefficient; S represents a state and X_S the state's coefficient or parameter for a given force or moment X. The flight test data was divided into *identification* and *validation* sets. The identification set is used to estimate the aerodynamic parameters; the validation set is used to verify if the estimated model is able to represent the aerodynamic forces and moments.

Under the previously mentioned premise, the *full* model was devised in the way that each of the aerodynamic forces (X, Y, Z) and moments (L, M, N) are a linear function of the following states (S): ϕ_r, θ_r, ψ_r, u, v, w, p, q, r, α, β, δ_f, δ_e, δ_r.

The *reduced* model presented in Eq. 4 was devised in order to test the states estimation capabilities of a simple and computational inexpensive linear model, that uses the states that can be measured onboard.

http://www.imavs.org/

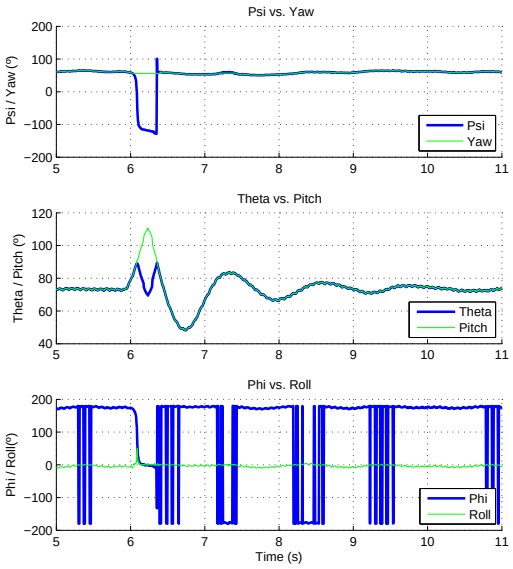


Figure 5: Euler angles (blue) and attitude angles (green).

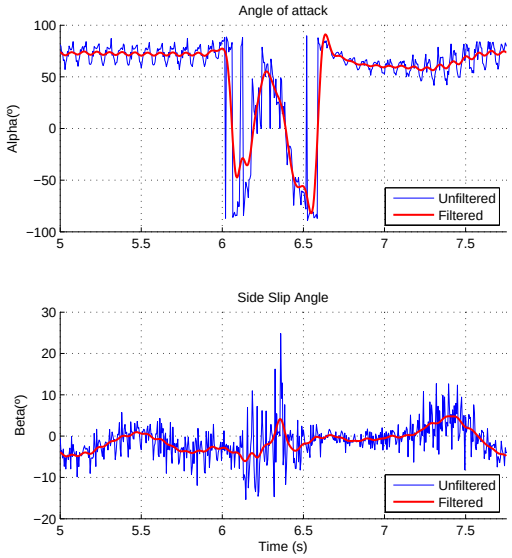


Figure 6: Attack (top) and side slip angles (bottom).

$$\begin{aligned} X &= X_0 + X_q q + X_\theta \theta + X_{\delta_e}\delta_e + X_{\delta_f}\delta_f \\ Y &= Y_0 + Y_p p + Y_\phi \phi + Y_{\delta_r}\delta_r + Y_{\delta_f}\delta_f \\ Z &= Z_0 + Z_q q + Z_\theta \theta + Z_{\delta_e}\delta_e + Z_{\delta_f}\delta_f \\ L &= L_0 + L_\theta \theta + L_{\delta_r}\delta_r + L_{\delta_f}\delta_f \\ M &= M_0 + M_\theta \theta + M_{\delta_e}\delta_e + M_{\delta_f}\delta_f \\ N &= N_0 + N_\theta \theta + N_{\delta_r}\delta_r + N_{\delta_f}\delta_f \end{aligned} \tag{4}$$

An ordinary least squares estimator was used to estimate the parameters in the linear models. As an example, to estimate the parameters in the equation for X in Eq. 4 we get:

$$\begin{bmatrix} \hat{X_0} \\ \hat{X_q} \\ \hat{X_\theta} \\ \hat{X_{\delta_e}} \\ \hat{X_{\delta_f}} \end{bmatrix} = (\mathbf{R}^\top \mathbf{R})^{-1} R^\top X \tag{5}$$

with

$$\mathbf{R} = \begin{bmatrix} 1 & q(1) & \theta(1) & \delta_e(1) & \delta_f(1) \\ 1 & q(2) & \theta(2) & \delta_e(2) & \delta_f(2) \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & q(N) & \theta(N) & \delta_e(N) & \delta_f(N) \end{bmatrix} \tag{6}$$

with the regression matrix $\mathbf{R}$ containing a total of N observations.

3.2.3 Model Validation

This subsection presents the aerodynamic forces and moments' estimation results for both the full and the reduced models.

The states that were used to compute the aerodynamic forces and moments were filtered (as mentioned in section 3.1.2). This is coherent with the current PID control solutions that have been previously implemented on the Delfly, as the high order fast oscillations are filtered from the accelerometer and gyroscope information, for controllability purposes.

http://www.imavs.org/

A set of different near-hover condition flight tests, with dissimilar elevator inputs was evaluated in this routine. The results show that the models are able to estimate the aerodynamic forces with great approximation. The moments, however, cannot be estimated as well as the forces but are still able follow the calculated moments (from the EOM) around the manoeuvres.

The results are presented for the same test and for the same part of the flight as presented in Figures 3 to 6. Hence, Figures 7 and 8 present the graphical evolution of the filtered forces and moments respectively. These Figures present the final results of the *validation* part of the system identification. The blue lines represent forces and moments calculated using the aircraft EOM; the red lines depict the full linear model's evolution; whereas the green lines describe the reduced model's behaviour.

Both models are able to follow all aerodynamic forces with a good approximation capability (Figure 7). The variation in the results for the peak in the X force result from the fact that the X force that was computed from the EOM is highly affected by the velocity component in w, which drops to negative values when the Delfly flies inverted. Both models are able to estimate the Z force with a very good performance. The Y force does not vary considerably in the (longitudinal) manoeuvre, as no rudder input was present. Despite some punctual differences, the Y force is also well estimated by both models.

None of the models is able to completely follow the aerodynamic moments' evolution (Figure 8). For the case shown here, the most important moment (M) is also the best estimated one. However, the *full* model is not able to completely follow the moments around the manoeuvre and the *reduced* model presents a slight phase lag on the results. The L and N moments are not well estimated by none of the models for longitudinal manoeuvres. Nevertheless, these are able to follow the computed (EOM) L and N moments for lateral inputs on the rudder, where it is noticeable a coupling between all 3 moments – a rudder deflection of the Delfly induces a nose-down manoeuvre, due to the vertical arm of the rudder, with respect to the flappers centre of gravity.

The estimated aerodynamic moments present a cycle averaged behaviour similar to the ones computed out of the EOM, pointing to a possible application for onboard control, using online filtering of the states or a more refined linear model.

A quantitative measurement of the performance of the model is given by calculating Pearson's correlation coefficient between each of the model's estimations and the forces and moments that were computed from the EOM. The correlation coefficient captures how two signals vary with respect to their means and is defined as $\rho = \frac{\mathrm{cov}(X_{EOM}, X_{estim})}{\sigma(X_{EOM})\sigma(X_{estim})}$. The best performance would be the highest correlation of $\rho = 1$, while completely decorrelated signals would give $\rho = 0$. Table 2 shows the correlation coefficients for the full model and reduced model. It can be seen that the forces X, Y, and Z are well estimated with $\rho \in [0.85, 0.99]$. The moment predictions are still reasonably correlated for the *full* model ($\rho \in [0.39, 0.62]$), but only slightly correlated for the *reduced* model $\rho \in [0.14, 0.43]$.

Pearson's Correlation Coefficient

Forces and Moments	**X**	**Y**	**Z**	**L**	**M**	**N**
Full Model	0.88	0.97	0.99	0.39	0.62	0.46
Reduced Model	0.85	0.95	0.99	0.14	0.43	0.21

Table 2: Pearson's Correlation Coefficient between each of the linear models and the calculated forces and moments.

http://www.imavs.org/

http://www.imavs.org/

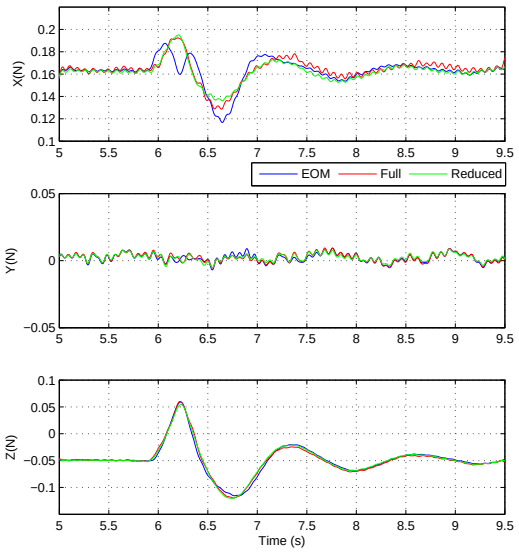


Figure 7: Calculated (blue) and estimated forces for both the full (red) and reduced (green) models, around the elevator input manoeuvre.

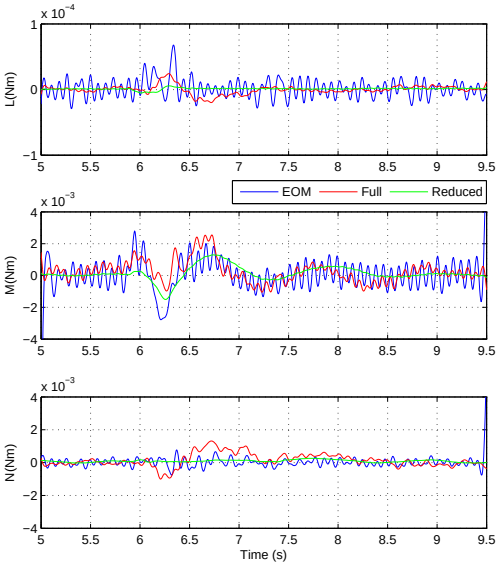


Figure 8: Calculated (blue) and estimated moments for both the full (red) and reduced (green) models, around the elevator input manoeuvre.

4 Conclusion

Two Delfly models were devised with the intention of achieving a linear time-invariant model description of the Delfly's aerodynamics. The linear models were designed from a control-driven perspective with the goal of using these models for onboard control and flight simulation purposes. Both the *full* and the *reduced* models were able to estimate the aerodynamic forces and moments, with better results on the *full* model. As a simple approximation, the consistency of the results conclude that the forces are well enough estimated to make a both linear models applicable for flight simulation as well as for onboard flight control. The moment's estimation results point to a possible application of these results for onboard control, using sensor fusion, onboard cycle average filtering and attitude thresholds. Further analyses will be conducted to search for other models, as well as to extend the results to other flight regimes and implement a controller based on the models to assess the flappers controllability.

References

[1] S. Ansari, R. Zbikowski, and K. Knowles. Aerodynamic modelling of insect-like flapping flight for micro air vehicles. *Progress in Aerospace Sciences*, 42(2):129–172, 2006.

[2] S. Baek and R. Fearing. Flight forces and altitude regulation of 12 gram i-bird. In *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 454–460, 2010.

[3] S. S. Baek. Autonomous ornithopter flight with sensor-based behavior. *Univ. California, Berkeley, Tech. Rep. UCB/EECS-2011-65*, 2011.

[4] H. Baruh. *Analytical Dynamics*. McGraw-Hill Higher Education, 1999.

[5] M. A. Bolender. Rigid multi-body equations-of-motion for flapping wing mavs using kanes equations. In *AIAA Guidance, Navigation, and Control Conference*, August 2009.

[6] J. Caetano, C. C. de Visser, B. Remes, C. de Wagter, and M. Mulder. Modeling a flappingwing mav: Flight path reconstruction of the delfly ii. AIAA, 2013.

[7] J. V. Caetano, C. C. de Visser, B. Remes, C. de Wagter, and M. Mulder. Controlled flight maneuvers of a flapping wing micro air vehicle: a step towards the delfly ii identification. In *AIAA Atmospheric Flight Mechanics Conference*, 2013.

[8] G. De Croon, K. De Clercq, R. Ruijsink, B. Remes, and C. De Wagter. Design, aerodynamics, and vision-based control of the delfly. *International Journal of Micro Air Vehicles*, 1(2):71–97, 2009.

[9] G. de Croon, E. de Weerdt, C. de Wagter, B. Remes, and R. Ruijsink. The appearance variation cue for obstacle avoidance. *IEEE Transactions on Robotics*, 28(2):529–534, 2012.

[10] G. de Croon, Groen, M., C. de Wagter, B. Remes, R. Ruijsink, and B. van Oudheusden. (accepted) design, aerodynamics, and autonomy of the delfly. *In Bioinspiration and Biomimetics*, 2012.

[11] Delfly Team. Delfly. www.delfly.nl, accessed on January 5th, 2013.

[12] M. Dickinson, F.-O. Lehmann, and S. Sane. Wing rotation and the aerodynamic basis of insect flight. *Science*, 284(5422):1954–1960, 1999.

[13] C. Ellington. The aerodynamics of hovering insect flight. i. the quasi-steady analysis. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 305(1122):1–15, 1984.

[14] C. Ellington, C. Berg, A. van den Willmott, and A. Thomas. Leading-edge vortices in insect flight. *Nature*, 384(19/26):626–630, 1996.

[15] B. Etkin and L. D. Reid. *Dynamics of Flight: Stability and Control*. Jonh Wiley & Sons, Inc., 3rd edition, 1996.

[16] J. Grauer, E. Ulrich, J. H. Jr., D. Pines, , and J. S. Humbert. Testing and system identification of an ornithopter in longitudinal flight. *Journal of Aircraft*, 48(2):660–667, March-April 2011.

[17] J. Grauer, E. Ulrich, J. H. Jr., D. Pines, and J. S. Humbert. System identification of an ornithopteraerodynamics model. AIAA, August 2010.

[18] M. Keennon, K. Klingebiel, H. Won, and A. Andriukov. Development of the nano hummingbird: A tailless flapping wing micro air vehicle. In *Proc. of 50th AIAA Aerospace Science Meeting, Nashville, TN, January*, pages 06–12, 2012.

[19] V. Klein and E. A. Morelli. *Aircraft System Identification: Theory And Practice*.

[20] K. Ma, P. Chirarattananon, S. Fuller, and R. Wood. Controlled flight of a biologically inspired, insect-scale robot. *Science*, 340(6132):603–607, 2013.

[21] V. Malolan, M. Dineshkumar, and V. Baskar. Design and development of flapping wing micro air vehicle. In *42nd AIAA Aerospace Sciences Meeting and Exhibit, 5 - 8 January, Reno, Nevada*, 2004.

[22] J. A. Mulder, Q. P. Chu, J. K. Sridhar, J. H. Breeman, and M. Laban. Non-linear aircraft flight path reconstruction review and new advances. *Progress in Aerospace Sciences*, 35:673–726, 1999.

[23] C. T. Orlowski and A. R. Girard. Modeling and simulation of nonlinear dynamics of flapping wing micro air vehicles. *AIAA journal*, 49(5):969–981, 2011.

[24] J. Reiner, G. J. Balas, and W. L. Garrard. Flight control using robust dynamic inversion and time-scale separation. *Automatica*, 32:1493–1504, 1996.

[25] H. E. Taha, M. R. Hajj, and A. H. Nayfeh. Flight dynamics and control of flapping-wing mavs: a review. *Nonlinear Dynamics*, 70:907–939, 2012.

[26] R. Wood. The first takeoff of a biologically-inspired at-scale robotic insect. *IEEE Tranactions on Robotics*, 24(2):341–347, 2008.

http://www.imavs.org/

Optimal hovering kinematics with respect to various flapping-wing shapes

Qi Wang[1], Hans Goosen[2] and Fred van Keulen[3]

[1] Delft University of Technology, Delft, The Netherlands
q.wang-3@tudelft.nl
[2] Delft University of Technology, Delft, The Netherlands
j.f.l.goosen@tudelft.nl
[3] Delft University of Technology, Delft, The Netherlands
a.vankeulen@tudelft.nl

http://www.imavs.org/

Abstract

Flapping-wing kinematics for insect flight has been studied for decades, especially since engineers became interested in flapping-wing micro air vehicles (FWMAVs). Previous work mainly focused on understanding kinematic patterns employed by different insects from the perspective of their trajectories and aerodynamics, and on optimization of kinematic parameters to enhance the understanding. However, systematic research on the impact of different wing shapes and corresponding kinematics is incomplete. In this paper, we search for energy-efficient kinematics for hovering flight for a series of wing shapes which are described by a Beta probability density function and the shape parameter $\hat{r}_1$ (the non-dimensional radius of the first moment of wing area) to guide the wing design for FWMAVs. Three kinematic patterns are considered in the optimization: (1) fully active and harmonic kinematics for rigid wings, (2) active kinematics with linear torsion from the wing root to wing tip, (3) kinematics with passive pitching motion. We found that for the first kinematic pattern more efficient hovering flight can be achieved by the wing shape with a larger $\hat{r}_1$, a smaller frequency and no heaving motion, and linearly torsional pitching leads to more energy-efficient flight compared with kinematics with constant pitching amplitude along the leading edge. Additionally, optimal hovering kinematics with passive pitching is accompanied by heaving motion irrespective of its wing shape, which is reflected in insect flight. Although it is important to generate sufficient lift force for hovering with passive kinematics, the presence of the heaving motion dramatically increases the energy consumption.

1 Introduction

As one of the most eminent fliers, insects deeply impress us with their ultralight body, agile and highly efficient flight as well as capabilities of hovering and low-speed forward flights. Insect wings play a crucial role in their excellent flight capabilities and have fascinated biologists and engineers for centuries. Development of high-speed video cameras makes it possible to observe the trajectory of a flapping-wing in detail. Lots of experimental data of flapping-wing kinematics was collected in this way. To better understand why insects adopt certain kinematics and to find more energy-efficient kinematics for FWMAVs, sinusoidal motion was assumed most frequently to simulate the flapping-wing kinematics in experimental numerical [1, 2] and analytical [3] studies of flapping-wing aerodynamics, especially for FWMAVs [4–6]. Optimization of the flapping-wing kinematics for given flight conditions was also widely conducted. For instance, Sane and Dickinson [7] conducted experiments to determine the optimal wing kinematics of the fruit fly, Berman and Wang [8] investigated energy-minimizing

kinematics for the given morphology of fruit fly, bumblebee and hawkmoth, and Khan and Agrawal [9] determined the optimal hovering kinematics for FWMAV by means of perturbing kinematic parameters in a sequence of experiments using a robotic flapping-wing device. Comparing those observed or optimized results for different insects with different morphologies, there exists a diversity of kinematic parameters, stroke plane pattern, flapping frequency and amplitude.

As the easily accessible aspect of insects wings, the wing morphology was studied systematically and thoroughly by several biologists [10–12]. A quarter-ellipse was widely used by researchers to represent the wing shape of insects like the fruit fly [13], bumblebee [14] and hawkmoth [8, 15]. In the paper of Ellington [10], the Beta probability density function (BPDF) was proposed to describe wing shapes based on experimental data of different insects' morphological parameters. However, no further study was done to explore the effect of wing shape on the optimal kinematics for hovering flight. In this work, a quasi-steady aerodynamic model [8] is used to estimate the aerodynamic loads and power requirement for hovering flight. To minimize the power consumption while producing sufficient lift force for hovering flight with harmonic kinematics and BPDF defined shapes, we employ a hybrid optimization method, i.e., a stochastic global optimization method based on the subset simulation algorithm [16, 17] for the rough solution and a gradient-based optimization method for the precise global solution.

This paper is organized as follows. In the second section, the morphological description of the wing model is given. In the third section, a aerodynamic model for calculating aerodynamic loads and power consumption and three related coordinate systems are introduced. In the fourth section, the detailed kinematics and optimization model are described. Three hovering kinematic patterns, including fully active flapping kinematics, kinematics with linear torsion along wing span and kinematics with passive pitching, are studied in the optimization. Finally, comparisons between optimal kinematics with different kinematic patterns are given to illustrate the effect of wing shape variation on the optimal hovering kinematics .

2 Wing Morphological Description

The morphology of flapping-wings plays an important role on generating aerodynamic forces and on flight control, especially the wing shapes and moments of inertia. Biologists have been trying to find the inherent law that governs the morphology of different insect wings for centuries, and a lot of work has been done on the data collection and analysis on the wing morphology. Among them, non-dimensional radii of the k^{th} moment of wing area were used to characterize the different wing shapes [10, 11], which is defined as,

$$\hat{r}_k = \left(\int_0^1 \hat{c}\hat{r}^k d\hat{r} \right)^{1/k}, \tag{1}$$

where $\hat{c} = c/\bar{c}$ and $\hat{r} = r/R$ are non-dimensional forms of chord length c and distance r from referenced wing chord to wing base, respectively, with $\bar{c}$ representing the average chord length and R the wing span. Ellington [10] studied wings from several species of insects, including Coleoptera, Diptera, Hymenoptera, Lepidoptera, Neuroptera and Odonata, and found that there exist strong relations between the non-dimensional radii of moments of wing area for a diverse collection of insects, even for some birds and bats. More specifically, for wings of different species studied by Ellington, $\hat{r}_1$ and $\hat{r}_2$ range from $0.4 \sim 0.6$ and $0.45 \sim 0.62$, respectively. Furthermore, $\hat{r}_2$ can be approximately expressed as a function of $\hat{r}_1$ for the same wing:

$$\hat{r}_2 = 0.929\hat{r}_1^{0.732}. \tag{2}$$

http://www.imavs.org/

In addition, Ellington [10] proposed to use the BPDF to describe the non-dimensionalized wing shape, as in,

$$\hat{c}(\hat{r};\alpha,\beta) = \frac{\hat{r}^{\alpha-1}(1-\hat{r})^{\beta-1}}{\int_0^1 \hat{r}^{\alpha-1}(1-\hat{r})^{\beta-1}d\hat{r}}, \qquad where, \quad \alpha = \hat{r}_1\frac{\hat{r}_1-\hat{r}_2^2}{\hat{r}_2^2-\hat{r}_1^2}, \beta = (\hat{r}_1-1)\frac{\hat{r}_1-\hat{r}_2^2}{\hat{r}_2^2-\hat{r}_1^2}. \tag{3}$$

Considering the relationship between $\hat{r}_1$ and $\hat{r}_2$ in Eq.(2), the chord length of an insect wing with the wing span R and average chord length $\bar{c}$ can be simply parameterized by a single parameter, i.e.,

$$c(r;\hat{r}_1) = \bar{c}\cdot\hat{c}\left(\frac{r}{R};\hat{r}_1\right). \tag{4}$$

The hawkmoth (Manduca sexta) is one of the most studied insects by biologists and referenced by man-made FWMAVs because of its moderate size and eminent hovering capability. In this work, these morphological parameters of hawkmoth wing [18], including the span length $R = 0.0519m$, aspect ratio $AR = 2.825$, wing mass $m_w = 5.79\times10^{-5}kg$ and moment of inertia of wing $I_\phi = 2.228\times10^{-8}kg\cdot m^2$, are used as basic parameters to construct wing shapes. Figure 1 shows the real wing shape of the hawkmoth and 5 different wing shapes described by the BPDF with $\hat{r}_1 = 0.40, 0.447, 0.50, 0.55, 0.60$. The wing shape for $\hat{r}_1 = 0.447$ coincides very well with real wing shape of hawkmoth, and the slight difference between them is due to the straight leading edge for the BPDF described wing shape.

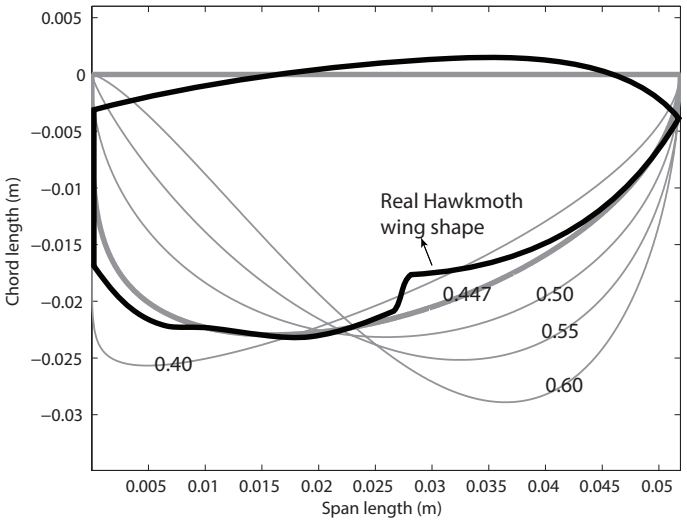


Figure 1: A series of wing shapes (gray) described by the BPDF with same R and $\bar{c}$ but different $\hat{r}_1$. A comparison is made between real hawkmoth wing shape (black) and wing shape described by the BPDF with $\hat{r}_1 = 0.447$ which was the measured value of hawkmoth wing by Willmott and Ellington [18].

Admittedly, there are limitations of the BPDF described wing shapes to completely duplicate some wing shapes because of straight leading edge and zero value at either end of the interval. Nonetheless, the wing with straight leading edge is easy to fabricate and hence a favorite of engineers, and two limitations mentioned above have negligible effects when using the quasi-steady model to calculate aerodynamic loads. Consequently, it is reasonable and convenient to use Eq.(4) to describe wing shapes with single shape parameter during the optimization.

http://www.imavs.org/

http://www.imavs.org/

3 Coordinate Systems and Aerodynamic Model

To optimize kinematics with different wing shapes described above, a quasi-steady aerodynamic model [8, 19, 20] is used to calculate aerodynamic loads. As the aerodynamic model uses the local pitching angles, velocities and accelerations of a wing strip to calculate the resultant loads, transformations are needed between this local coordinate system (CS) and the global CS. Therefore, these CSs related to transformations will be first described and those local information required for the calculation of aerodynamic loads will be derived as well before introducing the quasi-steady model.

3.1 Coordinate Systems

Assuming the wing is divided into many infinitesimal rigid strips, the kinematics of each strip can be described with three Euler angles, i.e. sweeping angle ϕ, heaving angle θ and pitching angle η. These angles and three involved CSs are illustrated in Figure 2. The global CS $x-y-z$ is fixed at the wing root, and its x axis coincides with the projection of the leading edge on the horizontal plane while the wing is at rest, the z axis is perpendicular to the stroke plane and upward, and the y axis is defined by the right-hand-rule. The co-rotating CS $x_c - y_c - z_c$ rotates with the wing, which is the result of three successive rotations around different axes. Three successive rotations can be denoted as three rotation matrices, i.e. $\mathbf{R}_\phi \rightarrow \mathbf{R}_\theta \rightarrow \mathbf{R}_\eta$:

$$\mathbf{R}_\phi = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R}_\theta = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}, \mathbf{R}_\eta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\eta & -\sin\eta \\ 0 & \sin\eta & \cos\eta \end{bmatrix}. \tag{5}$$

The angular velocity of the wing can be expressed as the sum of these successive angular velocity vectors, then its coordinates expressed in the co-rotating CS are obtained by transforming three angular velocity vectors to the co-rotating CS, i.e.,

$$[\omega_{x_c}, \omega_{y_c}, \omega_{z_c}]^T = \mathbf{R}_\eta^{\mathrm{T}} \mathbf{R}_\theta^{\mathrm{T}} \mathbf{R}_\phi^{\mathrm{T}} \left[0, 0, \dot{\phi}\right]^T + \mathbf{R}_\eta^{\mathrm{T}} \mathbf{R}_\theta^{\mathrm{T}} \left[0, \dot{\theta}, 0\right]^T + \mathbf{R}_\eta^{\mathrm{T}} \left[\dot{\eta}, 0, 0\right]^T. \tag{6}$$

One strip is chosen to illustrate the velocity and acceleration expressed in the local CS $x' - y' - z'$ at its geometric center P, as shown in Figure 2. First, the absolute velocity at P expressed in the co-rotating CS can be obtained like this,

$$[v_{x_c}, v_{y_c}, v_{z_c}]^T = [\omega_{x_c}, \omega_{y_c}, \omega_{z_c}]^T \times \left[r, 0, -\tfrac{1}{2} c(r; \hat{r}_1)\right]^T \tag{7}$$

where v_{z_c} and $-v_{y_c}$ are exactly velocity components $v_{x'}$ and $v_{y'}$ in the local CS. By differentiating is Eq.(7) with time, acceleration components at P expressed in the local CS are obtained as follows:

$$a_{x'} = -r(\ddot{\theta}\cos(\eta) - \dot{\theta}\dot{\eta}\sin(\eta) + \ddot{\phi}\cos\theta\sin\eta - \dot{\phi}\dot{\theta}\sin\theta\sin\eta + \dot{\phi}\dot{\eta}\cos\theta\cos\eta), \tag{8a}$$

$$a_{y'} = r(\ddot{\theta}\sin(\eta) + \dot{\theta}\dot{\eta}\cos(\eta) - \ddot{\phi}\cos\theta\cos\eta + \dot{\phi}\dot{\theta}\sin\theta\cos\eta + \dot{\phi}\dot{\eta}\cos\theta\sin\eta) - \tfrac{1}{2} c(r)(\ddot{\eta} + \ddot{\phi}\sin\theta + \dot{\phi}\dot{\theta}\cos\theta). \tag{8b}$$

3.2 Aerodynamic Model

When a flapping-wing is accelerated or decelerated, the added mass effect has to be considered for aerodynamics analysis. Referring to the added mass for thin rectangular plate [21], the added mass

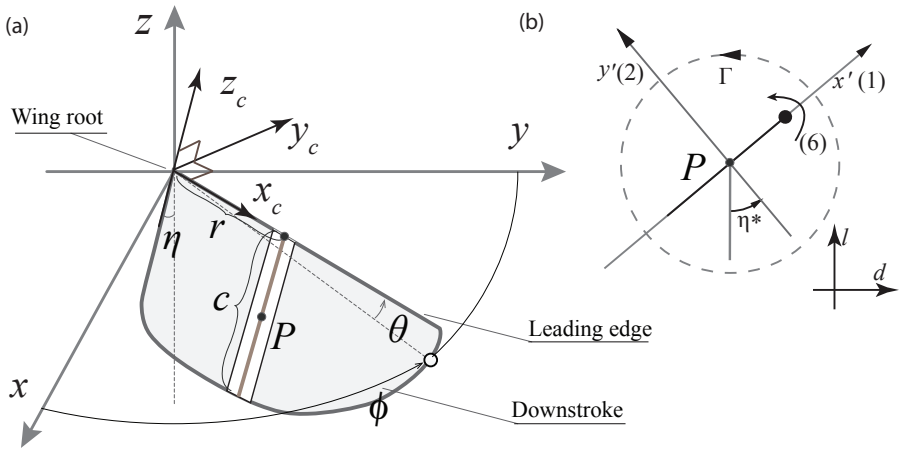


Figure 2: (a).Graph of three Euler angles and two related CSs. The gray shape represents the flapping-wing. P is the geometry center of the selected strip. The global CS, co-rotating CS are denoted as $x-y-z$ and $x_c-y_c-z_c$, respectively. It should be noted that both magnitudes of θ and η shown in the figure are negative according to the right-hand-rule. (b).Definitions of circulation (Γ), co-rotating CS $x-y-z$ and axes for added mass (modified from [19]).

tensor $\mathbf{M}$ per unit span length for the flapping wing is given as follows:

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{16} \\ m_{21} & m_{22} & m_{26} \\ m_{61} & m_{62} & m_{66} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\pi}{4}\rho_{air}c^2(r;\hat{r}_1) & 0 \\ 0 & 0 & \frac{9\pi}{128}\rho_{air}c^4(r;\hat{r}_1) \end{bmatrix}, \tag{9}$$

where ρ_{air} is the density of air ($1.29kg/m^3$), and m_{ij} represents the added mass induced load in the i^{th} direction on the wing due to a unit acceleration in the j^{th} direction of the wing. Subscripts "1" and "2" denote translation along x' and y' axes, and "6" denotes the rotation around the leading edge, as shown in Figure 2(b). Based on above definition, the two-dimensional quasi-steady aerodynamic model is used to calculate aerodynamic loads for the flapping-wing by integrating loads of each strip, as in,

$$F_{x'} = \int_0^R \left[-\rho_{air} v_{y'} \Gamma - m_{11} a_{x'} + \left(\frac{c(r;\hat{r}_1)}{\bar{c}R} m_w + m_{22} \right) v_{y'} \dot{\eta} - F^v_{x'} \right] dr, \tag{10a}$$

$$F_{y'} = \int_0^R \left[\rho_{air} v_{x'} \Gamma - m_{22} a_{y'} - \left(\frac{c(r;\hat{r}_1)}{\bar{c}R} m_w + m_{11} \right) v_{x'} \dot{\eta} - F^v_{y'} \right] dr, \tag{10b}$$

$$\tau_\eta = \int_0^R \left[\tfrac{1}{4} c(r;\hat{r}_1) \rho_{air} \sqrt{v^2_{x'} + v^2_{y'}} \Gamma - m_{66} \ddot{\eta} + (m_{11} - m_{22}) v_{x'} v_{y'} - \tau^v \right] dr, \tag{10c}$$

where the quasi-steady model divides aerodynamic loads into four terms: (1) loads for an inviscid flow, (2) inertial loads, (3) loads due to Coriolis effect, (4) viscous loads. It can be found that aerodynamic forces are decomposed into components in x' and y' directions for convenience since added mass in Eq.9 is defined in the local CS. Then, the lift force L and drag force D can be easily obtained by transforming $F_{y'}$ and $F_{x'}$ by the angle $\eta^\star$, as in Figure 2(b).

http://www.imavs.org/

To calculate the power consumption of hovering flight, the moments of inertia of the wings are required. Currently, we know the measured moment of inertia with sweeping motion I_ϕ for the wing shape with $\hat{r}_1 = 0.447$. If the mass is assumed to be uniformly distributed, the nominal moment of inertia I_ϕ^N can be calculated by $\int_0^R R^{-2} AR m_w c(r;\hat{r}_1) r^2 dr$. However, I_ϕ^N is larger than the measured I_ϕ due to inhomogeneous mass distribution for insect wings. To revise I_ϕ^N, I_θ^N and I_η^N for different wing shapes, a correction factor is introduced:

$$\lambda = \left(I_\phi / I_\phi^N\right)\big|_{\hat{r}_1=0.447}. \tag{11}$$

In addition, aerodynamic torques τ_ϕ and τ_θ can be calculated approximately by $\tau_\phi = sgn(\dot{\phi}) \int_0^R r dD$ and $\tau_\theta = -sgn(\dot{\theta}) \int_0^R r dL$. Referring to the definition in [8], the power consumption which is estimated by two terms: (1) power needed to perform rotation in a vacuum, (2) and the power used to overcome aerodynamic forces, is given in a mass-normalized form,

$$\bar{P}(t) = \Xi \left[\frac{1}{m_t} \sum_{i=\phi,\theta,\eta} \Omega_i \left(\lambda I_i^N \dot{\Omega}_i - \tau_i \right) \right], \tag{12}$$

where $\Xi[\cdot]$ is a predefined function whose value is 1 for positive value and 0 for non-positive, Ω_i denotes the angular velocity of corresponding rotation, and m_t is the total weight that wings have to bear (here we use the body weight of hawkmoth $m_t = 1.648 \times 10^{-3} kg$). For fully active kinematics, the whole flapping-wing system can not store energy according to the definition $\Xi[\cdot]$. Specially, for the passive pitching kinematics the power consumption is estimated by neglecting the term $\lambda I_\eta^N \eta \dot{\eta}$ representing the kinetic energy of pitching rotation.

4 Kinematics Description and Optimization Models

4.1 Kinematics Description

Based on the definition of Euler angles in section 3.1, sinusoidal motions are assumed for sweeping and heaving motions, as in,

$$\phi(t) = \phi_m \sin(2\pi f t), \tag{13a}$$

$$\theta(t) = \theta_m \sin(4\pi f t - \tfrac{1}{2} N\pi), \tag{13b}$$

which define the flapping motion with the figure of "∞" and "banana" shape of motion while N takes the value 0 and 1. Both shapes are commonly adopted by natural insects. For the pitching motion, three cases are considered: (1) active pitching with same amplitude along the leading edge, (2) wing torsion linearly increasing from the wing root to wing tip, (3) passive pitching mainly resulting from the combined effect of the flexible hinge at the wing root, inertia forces and aerodynamic forces.

4.1.1 Active Pitching

Active pitching here is simply defined as a sinusoidal motion, as in,

$$\eta(t) = \eta_m \sin(2\pi f t - \tfrac{\pi}{2}). \tag{14}$$

http://www.imavs.org/

4.1.2 Linearly Increased Pitching

According to numerical simulations [22] and experimental studies [15], we know flapping insect wings exhibit increasing torsion from the wing root to wing tip. We attempt to partly figure out the reason via studying the influence of torsion on the flight efficiency with different wing shapes by assuming a linear wing torsion along the leading edge, i.e.,

$$\eta_m(r) = k_\eta r + \eta_m^{root}, \qquad 0 \le k_\eta \le \frac{\pi}{2R}, \tag{15}$$

where η_m^{root} maintains the active kinematics as described in 4.1.1 at the root, the lower bound of the slope k_η denotes the case of a constant pitching amplitude along the leading edge, and the upper bound denotes another extreme case of $\pi/2$ shift of pitching amplitudes between the wing root and wing tip. This linear approximation is close to many observed torsional shapes and allows the torsion to be described by a single parameter.

4.1.3 Passive Pitching

Another characteristic of the insect flapping-wing is the passive pitching during the stroke reversal or even the whole stoke cycle. To mimic that, a rigid wing model with a spring at wing root and with the shape described by the BPDF is assumed. Lagrange equation is used to determine the passive pitching kinematics which is governed by the spring stiffness k and non-conservative forces imposed on the wing, as in,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\eta}}\right) - \frac{\partial L}{\partial \eta} = Q, \tag{16}$$

where $L\left(\phi, \dot{\phi}, \eta, \dot{\eta}; t\right)$ denotes the Lagrangian which consists of kinetic and potential energies of flapping-wing, and Q is the sum of generalized non-conservative forces which are related to components of the non-conservative forces in global CS and transformations between global and generalized coordinate (η), as in,

$$L = K - P = \frac{1}{2}\lambda I_\eta^N \dot{\eta}^2 - \frac{1}{2}k\eta^2, \tag{17}$$

$$Q = \int_0^R \frac{\partial x}{\partial \eta} d(f_I^x + f_A^x) + \frac{\partial y}{\partial \eta} d(f_I^y + f_A^y) + \frac{\partial z}{\partial \eta} d(f_I^z + f_A^z). \tag{18}$$

The global coordinates for the geometric center of each strip along the span-wise are

$$[x, y, z]^T = \mathbf{R}_\phi \mathbf{R}_\theta \mathbf{R}_\eta \left[r, 0, -\tfrac{1}{2}c(r; \hat{r}_1)\right]^T \tag{19}$$

and two non-conservative forces, i.e, inertial forces $\mathbf{f}_I$ and aerodynamic forces $\mathbf{f}_A$, can be obtained as follows:

$$d\mathbf{f}_A = \mathbf{R}_\phi [0, dD, dL]^T = -\sin\phi dD \mathbf{e}_x + \cos\phi dD \mathbf{e}_y + dL \mathbf{e}_z, \tag{20}$$

$$d\mathbf{f}_I = \mathbf{R}_\phi [0, -r\ddot{\phi} dm, r\ddot{\theta} dm]^T = r\ddot{\phi}\sin\phi dm \mathbf{e}_x - r\ddot{\phi}\cos\phi dm \mathbf{e}_y + r\ddot{\theta} dm \mathbf{e}_z, \tag{21}$$

where the dD and dL are the drag and lift forces of a strip along the leading edge, and $\mathbf{e}_x$, $\mathbf{e}_y$ and $\mathbf{e}_z$ denote three basic vectors of the global CS. Inserting Eqs. 17 and 18 into the Lagrange equation, we get:

$$\lambda I_\eta^N \ddot{\eta} + k\eta - \frac{1}{2}\int_0^R c(r; \hat{r}_1) dD \cos\eta - \frac{1}{2}\int_0^R c(r; \hat{r}_1) dL \cos\theta \sin\eta + \frac{m_w}{2\bar{c}R}\int_0^R r c^2(r; \hat{r}_1) dr \ddot{\phi} \cos\eta - \frac{m_w}{2\bar{c}R}\int_0^R r c^2(r; \hat{r}_1) dr \ddot{\theta} \cos\theta \sin\eta = 0. \tag{22}$$

http://www.imavs.org/

4.2 Optimization Model

Based on wing shapes described by the BPDF and analytic quasi-steady aerodynamic model, it is currently possible to conduct optimization by considering the effect of wing shapes on the optimal kinematics with different kinematic patterns. The general optimization model can be summarized as:

$$\begin{aligned} &\textbf{find} && f^*, \phi_m^*, \eta_m^*(k), \theta_m^* \\ &\textbf{minimize} && \hat{P} = \frac{1}{T}\int_T \bar{P}(t)dt \\ &\textbf{subject to} && \text{Harmonic kinematics and Wing shapes with various } \hat{r}_1. \\ & && \hat{L} = \frac{2}{Tm_tg}\int_T L(t)dt \geq 1 \\ & && 0 < f \leq 100,\ 0 < \phi_m \leq \tfrac{\pi}{2},\ 0 < \eta_m \leq \tfrac{\pi}{2},\ -\tfrac{\pi}{2} \leq \theta_m \leq \tfrac{\pi}{2} \end{aligned} \tag{23}$$

where averaged power consumption and lift force are used as objective and one of the constraints. 21 discrete values for the wing shape parameter $\hat{r}_1$ are chose from the meaningful range $0.4 \sim 0.6$ with a same interval, and independent optimization is conducted for each instance. Since kinematic optimization problems are normally nonlinear and non-convex [22], a stochastic global optimization method based on the subset simulation algorithm (SubSim optimization) [16, 17] is first used to identify the small domain where the global optimum is located. Then, the gradient-based active-set optimization algorithm by calling the *fmincon* function of Matlab® is used to find the global optimal results.

5 Results and Discussion

5.1 Active Flapping Kinematics

In order to find the most efficient active kinematics with different wing shapes both for hovering in "∞" and "banana" kinematic patterns, the SubSim optimization is used first to determine the rough global solution for all wing shapes, as shown by the four types of points in Figure 3. Then, based on results from the SubSim optimization, initial points for the gradient-based optimization are selected within $85° \sim 90°$, $50° \sim 60°$, $-3° \sim 3°$ and $(10 \sim 20)Hz$ for sweeping, pitching and heaving amplitudes, and frequency, respectively.

It can be found from Figure 3, that a more efficient hovering flight can be achieved by a wing shape with a larger $\hat{r}_1$ and a smaller frequency. For different wing shapes the optimal kinematics always correspond to maximal sweeping motions (90°), minimal heaving motions (0°) and moderate pitching motions ($50° \sim 60°$).

5.2 Torsional Wing Kinematics

To compare the optimal kinematics for torsional wing with results in Section 5.1, various wing shapes are explored to search for the most efficient f, ϕ_m, η_m^{root} and k_η. Results are given in Figure 4. Figure (a) indicates that the flapping flight with linearly torsional wing is more efficient than with constant pitching amplitude irrespective of its shape, which partly explains why insects with various wing shapes use flexible wing to generate wing torsion during flapping flight. For the torsional wing, it can be seen from Figure (b) that the optimal pitching amplitude shift between the wing root and wing tip shrinks with the increasing of $\hat{r}_1$. We attribute this to the compromise between lift generation and

http://www.imavs.org/

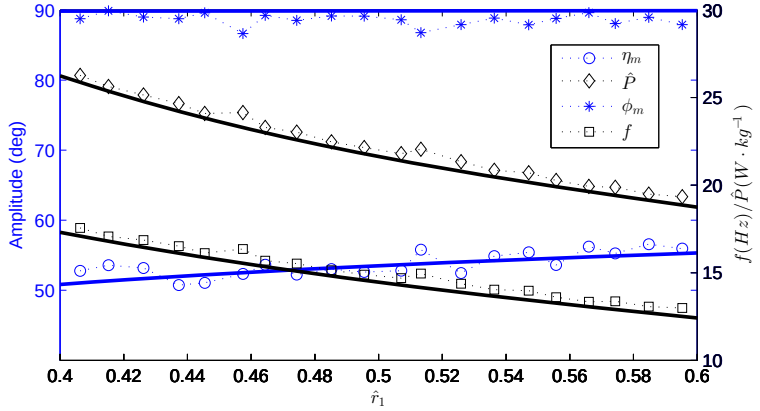


Figure 3: Optimal results for active flapping-wing kinematics for 21 wing shapes. Sets of different types of points represent the results of the SubSim optimization. Solid lines represent results of a gradient-based optimization based on those points.

energy consumption when the wing surface is more inclined to distribute at the wing tip for higher $\hat{r}_1$.

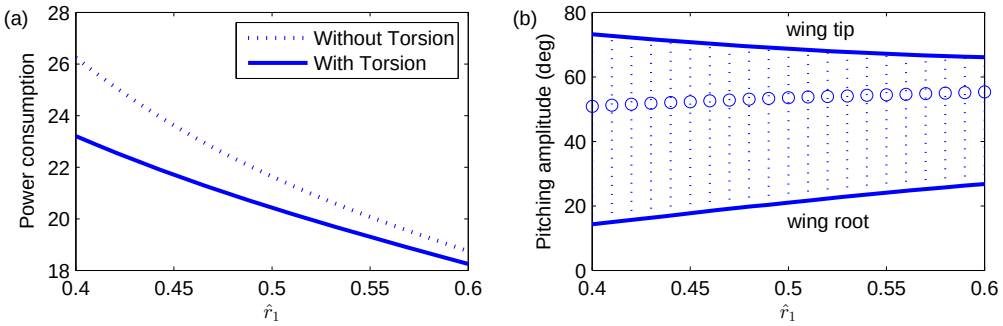


Figure 4: Comparison of optimal power consumption and pitching amplitudes between flapping flight with and without wing torsion while flapping in a horizontal stroke plane. Circles in (b) denote the optimal pitching amplitudes without wing torsion.

5.3 Passive Pitching Kinematics

Some research has been done to study contributions of inertial force and aerodynamic force to the passive pitching motion, and showed that the inertial force of the wing structure plays a more important impact on the passive pitching. Here, maintaining parameters $\phi_m = 60.3°$, $\theta_m = 3°$, $f = 26.3Hz$, $\hat{r}_1 = 0.447$ and $k = 8 \times 10^{-4} Nm/rad$, the numerical analysis based on the Lagrange equation is conducted for "∞" pattern kinematics. Passive pitching angles within first two periods are plotted in Figure 5, from which we can find the passive pitching amplitude due to inertial force is almost double that due to aerodynamic force. Furthermore, we plot results taking the variation of the wing shape

http://www.imavs.org/

into consideration in the Figure 6. The left figure shows that amplitudes due to both sources are proportional to the value of the non-dimensional radius of the first moment of wing area $\hat{r}_1$. The second figure indicates that contributions of inertial force and aerodynamic force remain almost constant for different wing shapes described by the BPDF.

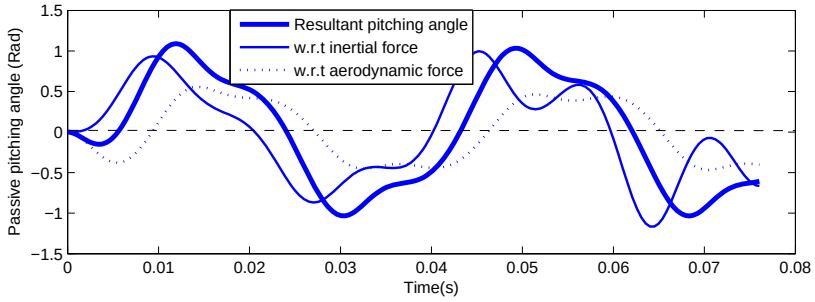


Figure 5: Contributions of inertia and aerodynamic forces to passive pitching angles in the first two cycles for "∞" pattern kinematics.

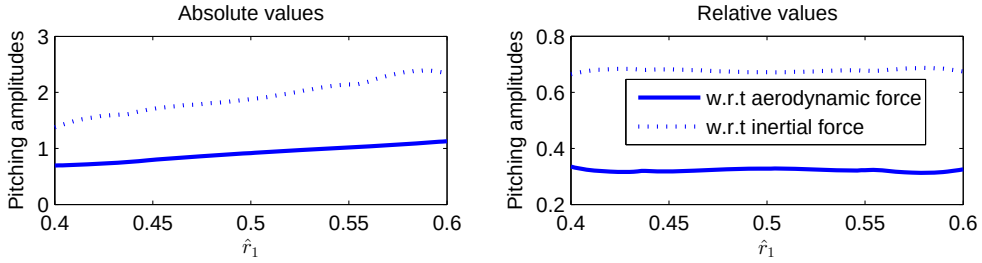


Figure 6: Contributions of inertia and aerodynamic forces to passive pitching amplitude with different wing shapes for "∞" pattern kinematics.

Due to the non-convexity of the problem and to decrease optimization time, the gradient-based optimization is not used in this case. Instead, the SubSim optimization method is used twice with relatively small number of samples to determine the global domain and to get the approximate global solution. First of all, the feasible domain for the design variables is reduced to $10 \sim 20Hz$, $4 \times 10^{-4} \sim 9 \times 10^{-4} Nm/rad$, $85° \sim 90°$ and $0° \sim 60°$ for f, k, ϕ_m and θ_m, respectively. Then, further optimization is performed with the SubSim optimization method again. As shown in Figure 7, it can be found that for wing shapes described with the BPDF the optimal spring stiffness is within $4 \times 10^{-4} \sim 6 \times 10^{-4} Nm/rad$, optimal frequency ranges from 12 to $15Hz$, and optimal heaving amplitude decreases from near 60° for $\hat{r}_1 = 0.4$ to near 0° for $\hat{r}_1 = 0.6$. The optimal sweeping amplitude is still $\pi/2$, the same as fully active optimal kinematics, which is not plotted. The passive pitching amplitude ranges from 80° to 60°, and the power consumption ranges from 80 to $30W/Kg$, both of which are larger than those for fully active optimal kinematics. The low efficiency of the passive pitching kinematics compared to active pitching kinematics can be explained by the introduction of the heaving motion, which is essential to generate enough lift force for passive pitching kinematics. On the other hand, it is worth mentioning that both the complexity and the total mass of the wing actuation mechanism are significantly reduced with passive pitching motion, and that the power required for

http://www.imavs.org/

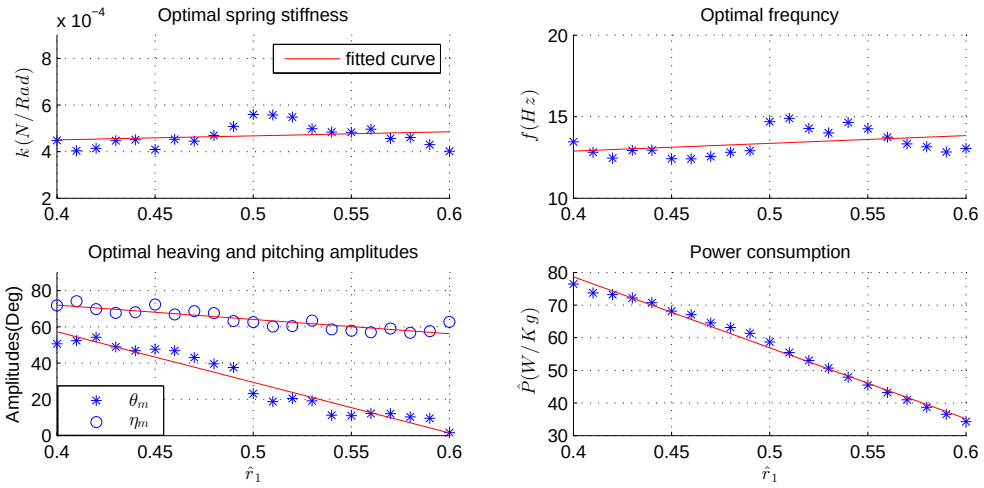


Figure 7: Optimal spring stiffness k, frequency f, heaving amplitude θ_m and corresponding passive pitching amplitude η_m and power consumption $\hat{P}$ for passive pitching kinematics with different wing shapes for "∞" pattern kinematics. These discrete results are fitted linearly to better reflect the trend.

active pitching was not considered.

6 Conclusions

Wing shapes described by the BPDF offer more options for FWMAVs compared to the quarter-ellipse wing shape or completely mimicking insect wings, and they are easy to fabricate due to the straight leading edge and smooth wing edges. From the analysis of optimization results applied to these wing shapes for hovering flight, some useful guidelines for wing design of FWMAVs are obtained. Firstly, as approximately harmonic motions are relatively easy to achieve by various actuation mechanisms especially for those based on crank-link mechanism, simply from the perspective of hovering flight, the fully actively actuated harmonic kinematics might be the most efficient scheme, while combined with a wing torsion along the span which can be achieved by deceasing the torsion stiffness. If passive flapping-wing pitching is adopted, kinematics with heaving motion is a better way to save energy. When considering the weight of the whole FWMAV which benefits from the simplicity of the actuation mechanism compared to a fully actuated mechanism, the "less" efficient passive pitching kinematics seems still favorable as a whole. Finally, for those FWMAVs mainly designed for hovering, wing shapes with larger $\hat{r}_1$ are recommended.

In the near future, more general kinematics might be considered for the optimization and the performance of kinematics with both passive pitching and torsion is also of great interest.

Acknowlegements

We would like to thank the financial support on my PhD project from China Scholarship Council and beneficial discussion with H.J. Peters.

http://www.imavs.org/

References

[1] H. Aono, Chang-kwon Kang, Carlos E. S. Cesnik, and Wei Shyy. A numerical framework for isotropic and anisotropic flexible flapping wing aerodynamics and aeroelasticity. In *28th AIAA Applied Aerodynamics Conference*, number July, pages 1–25, Chicago, Illinois, 2010.

[2] Attila J Bergou, Leif Ristroph, John Guckenheimer, Itai Cohen, and Z Jane Wang. Fruit flies modulate passive wing pitching to generate in-flight turns. pages 1–4, 2009.

[3] S. P. Sane and Michael H. Dickinson. The aerodynamic effects of wing rotation and a revised quasi-steady model of flapping flight. *J. Exp. Biol.*, 205(8):1087–1096, April 2002.

[4] CT Bolsman. *Flapping wing actuation using resonant compliant mechanisms, an insect inspired design.* Phd thesis, Delft University of Technology, 2010.

[5] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood. Controlled Flight of a Biologically Inspired, Insect-Scale Robot. *Science*, 340(6132):603–607, May 2013.

[6] JP Whitney. *Design and Performance of Insect-Scale Flapping-Wing Vehicles.* Phd thesis, Harvard University, 2012.

[7] S. P. Sane and Michael H. Dickinson. The control of flight force by a flapping wing: lift and drag production. *J. Exp. Biol.*, 204(15):2607–2626, August 2001.

[8] Gordon J. Berman and Z. Jane Wang. Energy-minimizing kinematics in hovering insect flight. *Journal of Fluid Mechanics*, 582:153, June 2007.

[9] Zaeem a. Khan and Sunil K. Agrawal. Optimal Hovering Kinematics of Flapping Wings for Micro Air Vehicles. *AIAA Journal*, 49(2):257–268, February 2011.

[10] C. P. Ellington. The Aerodynamics of Hovering Insect Flight. II. Morphological Parameters. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 305(1122):17–40, February 1984.

[11] T Weis-Fogh. Quick Estimates of Flight Fitness in Hovering Animals, Including Novel Mechanisms for Lift Production. *J. Exp. Biol.*, 59(1):169–230, August 1973.

[12] Robin J Wootton. Functional morphology of insect wings. *Annu. Rev. Entomol.*, 37:113–140, 1992.

[13] A. R. Ennos. Inertial and aerodynamic torques on the wings of diptera in flight. *J. Exp. Biol.*, 142(1):87–95, March 1989.

[14] R. Dudley and C. P. Ellington. Mechanics of Forward Flight in Bumblebees: I. Kinematics and Morphology. *J. Exp. Biol.*, 148(1):19–52, January 1990.

[15] AP Willmott and CP Ellington. The mechanics of flight in the hawkmoth Manduca sexta. I. Kinematics of hovering and forward flight. *J. Exp. Biol.*, 200(21):2705–2722, November 1997.

[16] Hong-Shuang Li and Siu-Kiu Au. Design optimization using Subset Simulation algorithm. *Structural Safety*, 32(6):384–392, November 2010.

[17] Qi Wang, ZhenZhou Lu, and ZhangChun Tang. A novel global optimization method of truss topology. *Science China Technological Sciences*, 54(10):2723–2729, August 2011.

[18] AP Willmott and CP Ellington. The mechanics of flight in the hawkmoth Manduca sexta. II. Aerodynamic consequences of kinematic and morphological variation. *J. Exp. Biol.*, 200(21):2723–2745, November 1997.

[19] A. Andersen, U. Pesavento, and Z. Jane Wang. Unsteady aerodynamics of fluttering and tumbling plates. *Journal of Fluid Mechanics*, 541:65–90, October 2005.

[20] Umberto Pesavento and Z. Jane Wang. Falling Paper: Navier-Stokes Solutions, Model of Fluid Forces, and Center of Mass Elevation. *Physical Review Letters*, 93(14):144501, September 2004.

[21] C.E. Brennen. A review of added mass and fluid inertial forces. Technical report, Naval Civil Engineering Laboratory, Port Hueneme, California, 1982.

[22] HJ Peters, JFL Goosen, and Fred Van Keulen. Flapping wing performance related to wing planform and wing kinematics. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM*, volume 31, pages 1–11, Indianapolis, Indiana.

http://www.imavs.org/

Scientific Session 4

MAV systems, navigation and control

Chairman : Simon Lacroix

- Jan Bolting, Francois Defay and Jean-Marc Moschetta. ***Differential GPS for small UAS using consumer-grade single-frequency receivers***
- Murat Bronz, Jean Philippe Condomines and Gautier Hattenberger. ***Development of an 18cm Micro Air Vehicle : QUARK***
- Guangying Jiang and Richard Voyles. ***Hexrotor UAV Platform Enabling Dextrous Aerial Mobile Manipulation***

http://www.imavs.org/

Differential GPS for small UAS using consumer-grade single-frequency receivers

Jan Bolting[1], Francois Defaÿ[2] and Jean-Marc Moschetta[2]

[1] University of Braunschweig - Institute of Technology, Braunschweig, Germany
j.bolting@tu-bs.de
[2] Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse, France
francois.defay@isae.fr, Jean-Marc.Moschetta@isae.fr

http://www.imavs.org/

Abstract

Consumer-grade single-frequency GPS receivers with their known limitations are the predominant means of localization for small Unmanned Aircraft Systems (UAS). More intricate maneuvers such as automatic landings require a level of accuracy this class of receivers does not provide. As a contribution to improve the positioning accuracy without sacrificing the low-cost approach of this class of vehicles, a Local Area Augmentation System (LAAS) has been developed, based on consumer-grade single-frequency miniature GPS receivers both for the base station and airborne positioning.

On the part of the airborne receiver, the conventional approach of carrier phase smoothing has been extended by incorporating Doppler measurements to propagate the position during carrier phase signal outages or in the event of cycle slips. Pseudoranges and the augmented carrier phase observations are merged by means of an indirect linear Kalman filter in the position domain. The characteristics of the error state allow for some simplifications that reduce the computing effort of the filter.

To evaluate the system's performance under dynamic conditions, raw GPS data have been collected on a ground based moving platform and processed with Simulink. The results show a significantly improved 3D position accuracy compared to the standalone receiver solution.

Contents

1 Introduction **2**

2 Observations of a GNSS receiver **2**

3 Base station **3**
3.1 Base station positioning 4

4 Airborne positioning algorithm **4**
4.1 Background Trajectory 5
4.2 Absolute positioning 6

5 Experimental results **7**
5.1 Base positioning 7
5.2 Dynamic tests on the ground 7

6 Conclusions and future work **10**

http://www.imavs.org/

1 Introduction

Progress in low-cost sensor and computing technology within the last years has led to a growing number of new low-cost UAS designs for civil applications. Typically, they rely on single-frequency GPS receivers for localization due to their low-cost profile. Despite the efforts of manufacturers to improve the accuracy of single-frequency receivers by better modeling of atmospheric effects and the availability of space based augmentation systems such as EGNOS, today's low cost receivers are limited to 3D positioning errors of the order of several meters ($1\,\sigma$). This is adequate for cruise flight navigation but is expected to be a severe constraint for enlarging the flight envelope of small UAS including more advanced maneuvers such as automatic takeoffs and landings in confined areas and formation flight of multiple UAS.
In civil and military aviation, Ground Based Augmentation Systems (GBAS) are used to enhance the accuracy of GNSS positioning to a level that, for instance, allows fully automatic approaches. Unfortunately, commercial differential GNSS receivers do not meet the cost requirements of small UAS for civil applications. There are however low-cost receivers on the market that output raw code, carrier phase and Doppler observations and are thus in principle suitable for differential positioning techniques. This has been the motivation to investigate the achievable accuracy of a ground based augmentation system for UAS based on low-cost consumer-grade receivers.
There have been efforts before to use inexpensive single-frequency receivers for kinematic precise positioning applications, mostly for positioning of ground vehicles, see e.g. [4], [7]. The authors of [9] use L1 phase observations collected with a u-blox LEA-4T for the reconstruction of relative flight trajectories and obtain positive results using time-differenced carrier phase observations.
Precise absolute positioning with low cost single-frequency receivers is challenging compared to dual-frequency navigation grade receivers that are routinely used for surveying applications. Their observations typically exhibit a higher noise level, degrading the accuracy of the code based position and thus enlarging the search space for carrier ambiguity resolution. Since the ionospheric range error cannot be eliminated by forming the ionosphere-free combination of the two signals, corrections are valid only for up to about 10 km. In addition, they lack the advantage of accelerated ambiguity resolution by means of the wide-lane combination of the two carrier phase observations.

Embedded systems on board of small UAS are typically based on sub - GHz microcontrollers due to power and size constraints, thus a positioning system that requires low computing power is desirable. Moreover, continuous availability of a position estimate is of the essence to ensure operational safety. This article describes an approach towards meeting these two requirements. Results are given of a first implementation of the augmentation system in Matlab/Simulink. It has been applied to raw data collected on a ground based moving platform.
The article is organized as follows: after introducing the observation model in section 2, section 3 presents the computation of broadcast corrections and the method that is proposed to obtain the reference station position, section 4 is devoted to the positioning algorithm of the airborne receiver. Finally, section 5 presents our experimental results and section 6 gives a short summary and an outlook on future work.

2 Observations of a GNSS receiver

A GNSS receiver's pseudorange, carrier phase and carrier Doppler shift observations are commonly modeled as

$$\rho_i = r_i + c(\Delta t_{rx} + \Delta t_{SV,i}) + \Delta r_{ion,i} + \Delta r_{trop,i} + \Delta r_{eph,i} + \Delta r_{MP,\rho,i} + \Delta r_{HW,\rho,i} \quad (1)$$

$$\Phi_i = r_i + c(\Delta t_{rx} - \Delta t_{SV,i}) - \Delta r_{ion,i} + \Delta r_{trop,i} + \Delta r_{eph,i} + \lambda_i N_i + \Delta r_{MP,\Phi,i} + \Delta r_{HW,\Phi,i} \quad (2)$$

$$v_{Do,i} = \dot{r}_{Rx,i} + c(\Delta \dot{t}_{rx} - \Delta \dot{t}_{SV,i}) - \Delta \dot{r}_{ion,i} + \Delta \dot{r}_{trop,i} + \Delta \dot{r}_{MP,\Phi,i} + \Delta \dot{r}_{eph,i} + \Delta v_{HW,Do,i} \quad (3)$$

with

$$\dot{r}_{Rx,i} = \dot{r}_i - \dot{r}_{SV,i}$$

where ρ_i is the pseudorange observation, r_i the slant range according to the broadcast ephemeris, c the nominal speed of light used by the GNSS system, Δt_{rx} the receiver clock offset, $\Delta t_{SV,i}$ the satellite clock offset, $\Delta r_{ion,i}$ the range error due to ionospheric diffraction, $\Delta r_{trop,i}$ the range error due to tropospheric diffraction, $\Delta r_{eph,i}$ the line of sight (LOS) broadcast ephemeris error, $\Delta r_{MP,\rho,i}$ the range error due to code multipath, Φ_i the carrier phase range observation, λ the nominal carrier phase wavelength, N the carrier cycle ambiguity, $\Delta r_{MP,\Phi,i}$ carrier phase multipath, $v_{Do,i}$ the Doppler shift observation, transformed to a velocity and $\dot{r}_{Rx,i}$ the slant range rate corrected for the satellite's LOS velocity. Hardware noise is represented by $\Delta r_{HW,\rho,i}$, $\Delta r_{HW,\Phi,i}$ and $\Delta v_{HW,Do,i}$ respectively. $\dot{r}_{SV,i}$ is the range rate due to satellite motion as computed from broadcast ephemeris.

Differential augmentation systems exploit the fact that a major part of the measurement errors is highly correlated between two receivers that are sufficiently close to one another, due to the fact that the signals they receive experience very similar conditions on their way through the atmosphere. This allows the estimation of the common errors once the position of one receiver - herein referred to as the base station - is known.

3 Base station

The base station consists of a GPS receiver that is placed at a known location and a data processing system that computes the corrections and broadcasts them via a wireless datalink to an unlimited amount of UAS.

Both code and phase observations evolve rapidly in time due to the high orbital velocity of the satellites (affecting r_i) and the rapidly drifting clock offset Δt_{rx} of the u-blox receiver's free running clock. While the latter is common to all observations and thus is eliminated by solving the navigation equation, the first imposes hard timing constraints on the data link and the synchronicity of the measurements of the base receiver and the airborne receiver when using a single-differencing approach. Therefore, r_i is computed from broadcast ephemerides (see [2]) and subtracted from the pseudorange observations to form estimates of the common range errors. The base receiver's clock error is not removed from the corrections to avoid increasing the variance of the corrections. Thus the **pseudorange corrections** are formed as

$$\rho_{c,i} = c(\Delta t_{rx} + \Delta t_{SV,i}) + \Delta r_{ion,i} + \Delta r_{trop,i} + \Delta r_{eph,i} + \Delta r_{MP,\rho,i} + \Delta r_{HW,\rho,i} \quad (4)$$

The phase observations are treated in the same way, forming **carrier phase corrections**:

$$\Phi_{c,i} = c(\Delta t_{rx} - \Delta t_{SV,i}) - \Delta r_{ion,i} + \Delta r_{trop,i} + \Delta r_{eph,i} + \lambda_i N_i + \Delta r_{MP,\Phi,i} + \Delta r_{HW,\Phi,i} \quad (5)$$

http://www.imavs.org/

Doppler corrections are obtained by time-differencing of carrier phase corrections.
Note that due to the fact that the broadcast ephemerides are only local approximations of the satellite orbits, it has to be made sure that the base station and the airborne receiver use the same set of ephemerides.

3.1 Base station positioning

It will seldom be the case that a surveyed reference point is available for the base station. Consequently, a way has to be found to estimate its position in a reasonable amount of time. An error of the base position directly translates into an additional position offset of receivers using its corrections, thus the quality of the ground station positioning determines the achievable accuracy for the UAS.
The International GNSS Service (IGS) provides observation data of a worldwide network of GNSS reference stations. A subset of these stations provide their data in close-to-real-time via NTRIP streams. Tests have been run and it could be demonstrated that, using the open-source RTK positioning software rtklib, the base receiver position can be computed with sufficient accuracy within 20 minutes, provided there is a reference station that is sufficiently close. Please see section 5.1 for experimental time-to-ambiguity-resolution results for three baselines up to 134 km.

4 Airborne positioning algorithm

The airborne component applies the received corrections to its own observations, canceling the major part of common errors, leading to the corrected observations

$$\rho_{i,c} = c\Delta t_{rx,c} + \Delta r_{MP,\rho,i} + \Delta r_{HW,\rho,i,c} + \epsilon\Delta r_{\rho,c} \tag{6}$$

$$\Phi_{i,c} = c\Delta t_{rx,c} + \lambda N_c + \Delta r_{MP,\Phi,i} + \Delta r_{HW,\Phi,i,c} + \epsilon\Delta r_{\Phi,c} \tag{7}$$

$$v_{Do,i,c} = c\Delta \dot{t}_{rx} + \epsilon\Delta \dot{r}_{\Phi,c} + \Delta v_{HW,Do,i,c} + \epsilon\Delta r_{Do,c} \tag{8}$$

where $\Delta t_{rx,c}$ is the difference of the clock offsets of base and airborne receiver, N_c the difference of their integer ambiguities, $\Delta r_{HW,\rho,i,c}, \Delta r_{HW,\Phi,i,c}$ and $\Delta v_{HW,Do,i,c}$ the combined noises of observations and corrections. Residual common errors due to timely and spatial de-correlation are represented by $\epsilon\Delta r_{\rho,c}$, $\epsilon\Delta r_{\Phi,c}$ and $\epsilon\Delta r_{Do,c}$.

The UAS position is obtained in two steps. First, a relative trajectory based on carrier phase time differences and integrated Doppler observations is computed. Second, the absolute position is obtained from pseudoranges and both are merged by an indirect Kalman Filter. Both steps are described in more detail in the following sections.

Position domain code smoothing is superior to range domain approaches under dynamic conditions for two major reasons: First, three coordinates to be estimated instead of up to e.g. about seven ranges for GPS leads to smaller filter matrices and thus reduces execution time.
Second, and more importantly, in the range domain, after loss of lock and reacquisition of a satellite, its smoothed range is lost and the respective filter channel is re-initialized. Afterwards, the smoothed range will be unusable while the filter re-converges. This can be impractical if less than 4 converged ranges are available, forcing the rover to use a range being still in the convergence phase and leading to a degraded position estimate.

On the other hand, in the position domain only an arbitrary set of four or more range and delta range observations has to be available at all times to keep up a continuous smoothed position. A reacquired satellite can be used immediately for positioning. This is a particularly valuable property

http://www.imavs.org/

in urban environments. Imagine an airborne rover that subsequently looses lock to all satellites for a few epochs, but manages to maintain a set of 4 satellites all the time. If the rover uses a range domain filter, it is left with a set of filtered ranges whose error is thrown back to that of their pseudorange measurements. A position domain filter might experience an increased noise level due to a higher PDOP associated with fewer satellites in view, but run continuously.

4.1 Background Trajectory

Numerous approaches exist to solve for the unknown integer ambiguities to obtain a highly accurate purely carrier phase based position. Most of them are search-based (e.g. LAMBDA, introduced by [8]), but there are promising analytical approaches for On-The-fly ambiguity resolution as well, e.g. [3]. There is no known technique that provides reliable and instantaneous ambiguity resolution for L1-only receivers though. The author of [5] shows, based on theoretical considerations, that at least 12 satellites in view are necessary for on-the-fly ambiguity resolution in 100% of all cases, a condition that is not met by the GPS constellation.
A simple alternative to exploit the excellent accuracy of carrier phase measurements without the need for ambiguity resolution consists in time-differencing. Since for a continuously tracked signal the ambiguities are constant in time, they are canceled by subtracting the observations of two subsequent epochs. The method has been successfully applied to standalone L1 phase observations for the analysis of flight maneuvers of aircrafts and birds (see [10], [11]).
As every technique involving phase observations, this method is susceptible to cycle slips and loss of the carrier signal. In situations where - e.g. due to steep maneuvers - the number of available phase observations drops below four, the accurately tracked trajectory is lost.
To overcome this drawback, the method has been extended by incorporating integrated Doppler observations that are less susceptible to dynamic stress, leading to better availability. Whenever the number of available carrier phase time differences drops below the minimum of four, they are augmented by integrated Doppler observations, i.e.

$$\delta\mathbf{r} = \begin{bmatrix} \delta r_{cp,1} \\ \vdots \\ \delta r_{cp,n_{cp}} \end{bmatrix} \quad \text{for } n_{cp} \geq 4 \tag{9}$$

$$\delta\mathbf{r} = \begin{bmatrix} \delta r_{cp,1} \\ \vdots \\ \delta r_{cp,n_{cp}} \\ \delta r_{Do,1} \\ \vdots \\ \delta r_{Do,(n_{cp}-4)} \end{bmatrix} \quad \text{for } n_{cp} < 4 \tag{10}$$

with

$$\delta r_{Do,i} = \int_{t_{k-1}}^{t_k} v_{Do,i}\,\mathrm{d}t \tag{11}$$

Where the vector $\delta\mathbf{r}$ holds the delta ranges used to compute the displacement between two epochs,

http://www.imavs.org/

$\delta r_{cp,i}$ are carrier phase time differences, $\delta r_{Do,i}$ are integrated Doppler delta ranges of satellites whose carrier phase time differences are not available and n_{cp} represents the number of available carrier phase time differences.

Doppler observations are a snapshot of the derivative of the trajectory, i.e. the velocity, at the time of measurement. Their integral therefore only approximates the true displacement and integrated Doppler ranges are noisier than carrier phase ranges. The advantage of not loosing track of the trajectory however largely outweighs this drawback, the more so as integrated Doppler observations are only used during the short periods of carrier phase outages.

The position increment between two epochs is calculated by linearizing the observation equation (here given for one range, note that $\delta r_{i,k}$ represents the carrier phase time differences augmented by integrated Doppler delta ranges)

$$\begin{aligned}\delta r_{i,k} = &\sqrt{(x_k^{SV} - x_k^{Rx})^2 + (y_k^{SV} - y_k^{Rx})^2 + (z_k^{SV} - z_k^{Rx})^2} \\ &-\sqrt{(x_{k-1}^{SV} - x_{k-1}^{Rx})^2 + (y_{k-1}^{SV} - y_{k-1}^{Rx})^2 + (z_{k-1}^{SV} - z_{k-1}^{Rx})^2} \\ &+c(\Delta t_{Rx,k} - \Delta t_{Rx,k-1}) + (\epsilon_k - \epsilon_{k-1})\end{aligned} \tag{12}$$

and by solving it iteratively for $\boldsymbol{x}_k$ by the method of least squares.

4.2 Absolute positioning

Based on the corrected pseudoranges, the absolute position is computed following the standard procedure of linearizing and iteratively solving the navigation equation. To correct for the signal propagation delay, pseudoranges are time-tagged at the receiver, as proposed by [1]. The resulting noisy position estimate is used to correct the position offset of the background trajectory by means of a simple linear Kalman filter.

The prediction-correction structure of a Kalman Filter usually requires a dynamic model of the system whose state is to be estimated and knowledge about the control inputs. In this case, the offset of the background trajectory from the true trajectory drifts very little over time thanks to its excellent relative precision. It can therefore be approximately modeled as being constant, i.e the dynamics matrix $\boldsymbol{A} \approx I_3$ which is equivalent to omitting the state prediction step.

This naturally leads to an indirect i.e. error state filter formulation, as the difference between the pseudorange trajectory and the background trajectory is a noisy measurement of the background trajectory offset. The filter equations become

$$\Delta \boldsymbol{x}_k = \boldsymbol{x_{b,k}} - \Delta \boldsymbol{x_{true,k}} \tag{13}$$
$$\Delta \hat{\boldsymbol{x}}_k^- = \boldsymbol{A}\boldsymbol{\Delta}\hat{\boldsymbol{x}}_{k-1} \tag{14}$$
$$\boldsymbol{P}_k^- = \boldsymbol{A}\boldsymbol{P}_{k-1}\boldsymbol{A}^T + \boldsymbol{Q} \tag{15}$$
$$\Delta \boldsymbol{z}_k = \boldsymbol{x}_{\rho,k} - \boldsymbol{x}_{b,k} \tag{16}$$
$$\boldsymbol{K}_k = \boldsymbol{P}_k^-(\boldsymbol{P}_k^- + \boldsymbol{R})^{-1} \tag{17}$$
$$\Delta \hat{\boldsymbol{x}}_k = \Delta \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\Delta \hat{\boldsymbol{x}}_k - \Delta \boldsymbol{z}_k) \tag{18}$$
$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k)\boldsymbol{P}_k^- \tag{19}$$

http://www.imavs.org/

$\Delta\boldsymbol{x}_k$ being the error state, $\Delta\boldsymbol{z}_k$ being the error state measurement, $\boldsymbol{K}_k$ being the filter gain, $\Delta\hat{\boldsymbol{x}}_k^-$, $\Delta\hat{\boldsymbol{x}}_k$ being the *a priori* and *a posteriori* error state estimate respectively, $\boldsymbol{P}_k^-$, $\boldsymbol{P}_k$ being the *a priori* and *a posteriori* estimation covariance matrix respectively, $\boldsymbol{Q}$ being the process noise and $\boldsymbol{R}$ being the measurement noise.

5 Experimental results

The system has been implemented in Matlab/Simulink and applied to observations collected on a rotating test bench on the ground. Flight tests have been performed as well but are not considered in this article since no RTK trajectory could be obtained in post processing. Base positioning using freely available close-to-real-time observations by the IGS network ([6]) has been tested for three baseline lengths up to 134 km.

Hardware Data have been collected with GPS receivers of type u-blox LEA-6T. The LEA-6T chipset is one of the least expensive on the market providing raw observations on the GPS L1 band. Receiver modules integrating the same receiver chip as the evaluation kit used for our experiments are available with a weight of 16g including an active antenna, making them suitable for small UAS down to a few hundred grams TOW. The rover and base antennas where an active patch antenna (u-blox ANN-MS-0-005) and a survey grade antenna on a surveyed reference position, respectively.

5.1 Base positioning

To verify the assumption that the position of the base receiver can be determined in a reasonable amount of time using freely available reference observations and open source software, tests for three successively large baselines have been performed. Data have been collected for 20 minutes and processed with rtklib. The positions obtained by averaging all positions with fixed ambiguities are of centimeter-level quality (see Table 1), better accuracy can be expected for longer sampling periods. Longer baselines have been tested but did not lead to fixed ambiguities within 20 minutes.

b [km]	t_{fix} [s]	n_{fix}[%]	σ_N [mm]	σ_E [mm]	σ_D [mm]	μ_N [mm]	μ_E [mm]	μ_D [mm]
8	403	68	4.2	3.7	9.4	2.2	−1.4	−19.9
64	96	9	1.8	3.6	10.6	0.8	1.3	1.7
134	187	53	12.2	11.5	22.6	−4.4	13.2	8.7

Table 1: LEA-6T base positioning with rtklib for different baselines (stations TLMF, PAYR, NARB). t_{fix} is the time when ambiguities where fixed for the first time, n_{fix} the percentage of epochs for which ambiguities could be fixed within 20 minutes.

5.2 Dynamic tests on the ground

To emulate flight conditions, the receiver has been placed on a freely rotating turntable with a radius of 4.2 m. As with all localization systems, a key challenge consists of the comparison of measurements from the system to be tested versus a known truth source. In this case the true trajectory could be reconstructed with RTK-level accuracy using the open-source software rtklib (see e.g. [7]). The bar

http://www.imavs.org/

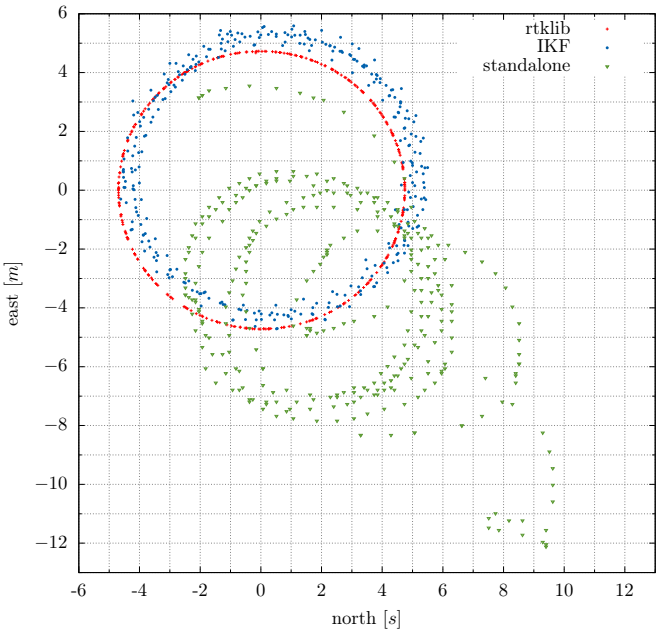


Figure 1: North and east coordinate of trajectory in local NED frame

http://www.imavs.org/

was rotated by one person. A second receiver was set up to record raw data at the campus' surveyed reference antenna. In parallel to the raw observations, the standalone position solution of the receiver has been logged. The test site was located 40 to 80 m from two two-story buildings, thus the rover receiver was likely to experience multipath.

The position errors over time are given in Fig. 2, see Fig. 3 and Tab. 2 for error statistics.
While not exceeding 5 m most of the time, the standalone error rises to up to 16 m in the east coordinate around 330 s. The filtered differential solution error stays well close to 1 m most of the time and exhibits a smaller variance. Note also that the filtered solution shows no sign of the two discontinuities at 25 s and 305 s in the receiver's solution. Regarding the total error (norm of the error vector), for the standalone solution 95 % of all 3D errors are smaller than 11.15 m. The Kalman filtered differential solution brings the 95% error range down to 1.35 m.

	σ_N [m]	σ_E [m]	σ_D [m]	μ_N [m]	μ_E [m]	μ_D [m]	HDOP	VDOP
Circular trajectory, baseline 180 m								
standalone	2.18	2.06	1.74	2.67	−4.41	0.74	-	-
Differential C/A	0.86	0.65	2.39	0.06	−0.2	−0.47	1.2-1.4	2.1-3.6
IKF	0.21	0.21	0.26	0.38	0.49	−0.85	1.2-1.4	2.1-3.6

Table 2: Position error standard deviations σ and means μ under static and dynamic conditions (300 s, 1 Hz)

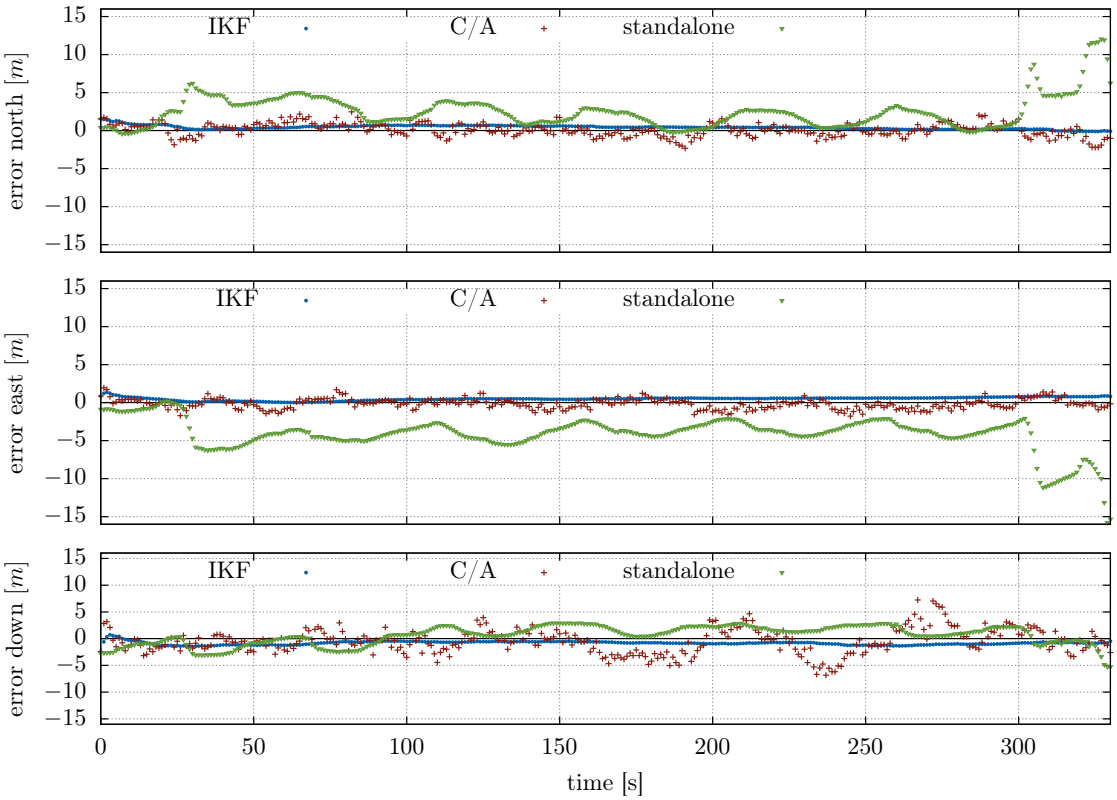


Figure 2: Position offset to RTK trajectory in local NED system: indirect Kalman Filter, differential C/A, LEA-6T standalone solution (SBAS enabled, configured for airborne platforms ≤ 4 g)

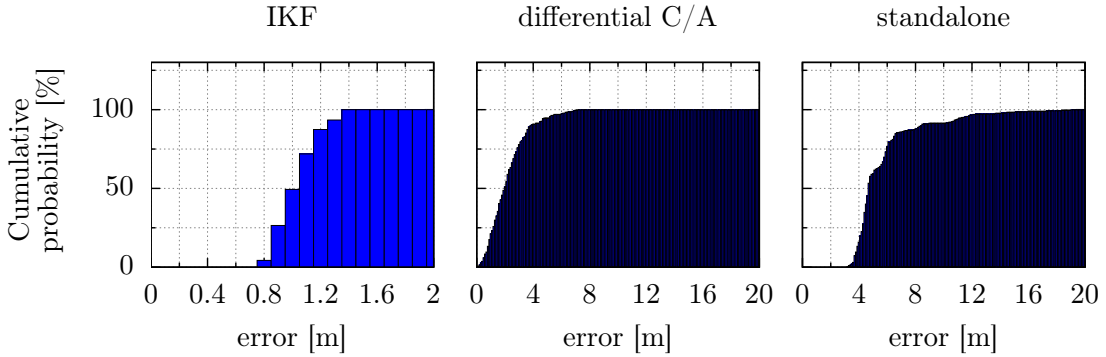


Figure 3: Cumulative distributions of 3D error on circular trajectory. Note the different scales of the x axes.

http://www.imavs.org/

6 Conclusions and future work

This work has investigated the feasibility of a local GPS augmentation system based on low cost hardware components. A code smoothing filter has been developed that takes advantage of the complementary qualities of carrier phase, Doppler and pseudorange observations, based on an indirect linear Kalman filter. The positioning filter has been implemented in Matlab/Simulink and applied to observation data collected on a rotating test bench. A highly accurate reference trajectory could be reconstructed using open - source RTK post-processing software. The results are quite promising and reveal the shortcomings of standalone dynamic GPS positioning. Differential positioning shows an improvement of the mean position error (the only exception being the vertical error whose mean values are of similar magnitude) as well as the error variance by roughly one magnitude. Further efforts will be focused on the implementation of the proposed system on an embedded system and the verification of its performance under flight conditions, including the tracking of the reference trajectory by means of GPS-independent instruments such as theodolites to exclude the possibility of common errors. Its robustness thanks to the integration of Doppler measurements makes the proposed system an excellent basis and safety fall-back for the evaluation of more accurate but less reliable techniques involving ambiguity resolution.

http://www.imavs.org/

References

[1] Neil Ashby and Marc Weiss. NIST Technical Note 1385 "Global Position System Receivers and Relativity". Technical report, United States National Institute of Standards and Technology, 1999.

[2] Interface Control Working Group. *Global Positioning System Interface Specification IS-GPS-200G*. Global Positioning Systems Directorate - Systems Engineering & Integration, September 2012.

[3] Xiaogang Gu and A. Lipp. DGPS positioning using carrier phase for precision navigation. In *Position Location and Navigation Symposium, 1994., IEEE*, pages 410 –417, apr 1994.

[4] Kjeld Jensen, Morten Larsen, Tom Simonsen, and Rasmus N. Jørgensen. Evaluating the performance of a low-cost gps in precision agriculture applications. In *RHEA-2012 - First International Conference on Robotics and Associated High-Technologies and Equipment for Agriculture*, 2012.

[5] Dennis Milbert. Influence of Pseudorange Accuracy on Phase Ambiguity Resolution in Various GPS Modernization Scenarios. *NAVIGATION*, 52:29–38, 2005.

[6] International GNSS Service. http://igscb.jpl.nasa.gov/network/netindex.html.

[7] Tomoji Takasu and Akio Yasuda. Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. *International Symposium on GPS/GNSS*, 2009.

[8] P. J G Teunissen. A new method for fast carrier phase ambiguity estimation. In *Position Location and Navigation Symposium, 1994, IEEE*, pages 562–573, 1994.

[9] J. Traugott, G. Dell'Omo, A.L. Vyssotski, D. Odijk, and G. Sachs. A time-relative approach for precise positioning with a miniaturized L1 GPS logger. In *Proceedings of ION GNSS 21th International Technical Meeting of the Satellite Division*, pages 1883–1894. ION GNSS, 2008.

[10] Johannes Traugott, Dennis Odijk, Oliver Montenbruck, Gottfried Sachs, and Christian Tiberius. Making a Difference with GPS. *GPS World*, May 2008:48–57, 2008.

[11] Johannes P. Traugott. *Precise Flight Trajectory Reconstruction Based on Time-Differential GNSS Carrier Phase Processing*. PhD thesis, Technische Unversitaet Muenchen, 2011.

Development of an 18cm Micro Air Vehicle : QUARK

Murat Bronz*, Jean-Philippe Condomines† and Gautier Hattenberger‡

l'Ecole National de l'Aviation Civile, Toulouse, 31055, France
murat.bronz@enac.fr, jean-philippe.condomines@recherche.enac.fr, gautier.hattenberger@enac.fr

Abstract

This paper describes the development of the **QUARK** micro unmanned air vehicle. The main objective of the study is to show the feasibility of designing an autonomous MAV smaller than 20 *cm* by using open-source programs and off-the-shelf components. Effort is given to show how to choose the system components and the design variables correctly in order to end up with an optimum design. An open-source multi-disciplinary conceptual aircraft design program called *CDSGN* is used in order to select the appropriate wing planform and the optimum battery capacity. Propeller and electric motor are selected throughout an optimization procedure, which analyze the performance of each suitable off-the-shelf propeller and motor combination for the specific mission requirements of the new design. A $2\,cm \times 2\,cm$ custom autopilot board is specifically designed according to the requirements of the **QUARK**. The modelization of the MEMS sensors (Micro-Electromechanical Systems), that are used on the autopilot, is described and analysed in order to improve the stabilisation and the navigation of the aircraft. Longitdinal dynamics of the vehicle are examined by using an innovative technique, and following that, corresponding flight control gains are tunned prior to the outdoor flight test.

The first prototype has been flown and a smaller 15.9 *cm* version is designed according to the obtained results. The expected endurance for the final version is over 35 *minutes* with reaching up to 35 *km* of range.

http://www.imavs.org/

1 Introduction

The birth of Micro Air Vehicles goes back to early 1990s, when the RAND Corporation performed a feasibility study. Following that interest, in 1996 Aerovironment showed the feasibility of flying a six-inch MAV in 1996, and later improved the performance of the existing design and performed a 16 minutes of flight in 1997 by manual control. After this achievement, Aerovironment was awarded a DARPA founded Phase II Small Business Innovation Research (SBIR) contract. As a result of this contract, the Black Widow MAV was designed, which is still one of the smallest fixed-wing MAV. Black Widow MAV had 30 minutes of endurance and was able to transmit real-time video to the ground within 1.8 *km* range. There was an autopilot for stability augmentation, however there was no GPS on-board for autonomous navigation. Recent progresses in the miniaturization of the inertial instruments i.e. Inertial Navigation System (INS) and Global Positioning System (GPS) hardware have led to the design of small, low-cost integrated navigation systems. Despite all these improvements, in the year of 2013, there is no fixed-wing MAV with autonomous navigation capability, which remains a big shortfall. Figure 1 shows INS relative strap-down system cost "projections" as a function of inertial instrument technology and performance.

*Doctor, Assistant-professor in Applied Aerodynamics, URI-Drones
†PhD Candidate in Automatic Control, MAIAA
‡Doctor, Assistant-professor in Flight Dynamics, MAIAA

The performance of MEMS-based gyroscopes and accelerometers is significantly affected by complex error characteristics that are stochastics in nature. Therefore, the modelisation of the sensors remains a challenge in order to improve the navigation.

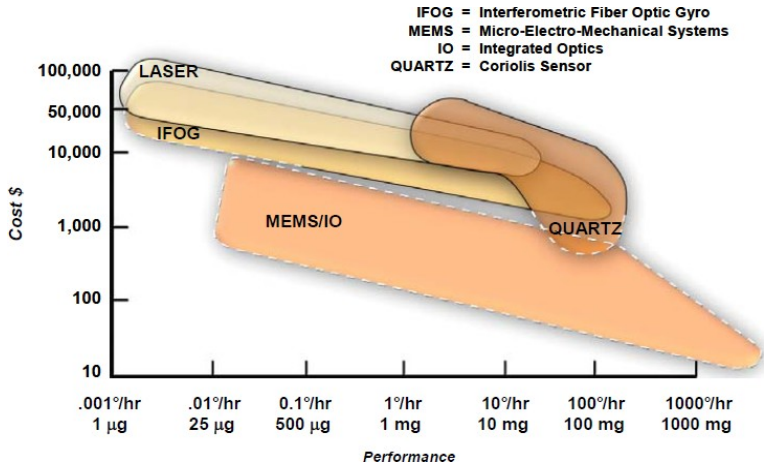


Figure 1: Strapdown INS cost as a function of instrument technology

On the rotary-wing MAVs, the Black Hornet MAV from Proxy Dynamics, demonstrated the state of the art for MAVs with autonomous navigation capability. The current study presents the development of the **QUARK** MAV, starting with the parametric and multi-disciplinary conceptual design phase in cooperation with propulsion system optimization.

2 Design Methodology

The design method was kept simple. The idea was to gather the most appropriate off-the-shelf components that are required for the flight, control, and communication of the aircraft and decide the minimum core size while taking into account the interference and center of gravity issues. Then by using a multi-disciplinary parametric conceptual design program called *CDSGN* [2], the wing planform, battery capacity, cruise speed, motor and propeller combination was optimized while the cost function being the endurance and range performance maximisation. Certain stability derivatives, such as pitch damping (C_{m_q}) and spiral stability ($C_{l_b}C_{n_r}/C_{l_r}C_{n_b}$), are used as constraints during the optimization phase.

3 Off-the-Shelf Component Search

One of the main objectives of this study was to use completely off-the-shelf commercially available components for the realization of the aircraft.

Mini-micro Scale Propellers. Among the available mini-micro scale propellers, seven different candidates were found and taken into consideration for the final propulsion system selection. Figure 2 shows the pictures and Table 1 presents the specifications of the candidate propellers. Each of the propellers modelled geometrically by measuring their chord length, and twist angle at several sections

http://www.imavs.org/

throhout the blade span. The corresponding airfoil geometry is extracted by cutting the propeller blade into pieces and then aerodynamic characteristics of each section airfoil is examined by using XFOIL[3]. Following that, the performance modelisation of the propellers done by using QPROP[4].



Figure 2: Pictures of candidtae commercially available off-the-shelf mini and micro propellers

Propeller Name	Blade number [-]	Diameter [inch]	Pitch [inch]	Mass [g]
Three Blade Prop	3	3.8	-	1.9
Graupner CAM Slim	2	3	3	0.8
Graupner CAM Speed	2	4	3	0.95
GEMFAN mini 3x2	2	3	2	1.0
GEMFAN mini 4x2.5	2	4	2.5	1.4
Red propeller	2	3	-	0.7
HobbyKing nano prop	2	2.2	-	0.3

Table 1: Coefficients of candidate commercially available off-the-shelf mini and micro electric motors

Micro Electric Motors. Likewise a micro electric motor database was created from the commercially available off-the-shelf motors. Figure 3 shows the candidate electric motors. A first order electric model is used to characterize the behavior of the motors. Only the internal resistance ($\mathcal{R}_m$), no-load current (I_0), and speed constant (K_v) are used for characterization. Table 2 presents the mass of each motor with their K_v coefficient.

4 Design Parametrization

Once the required electronics was defined, the sizing of the aircraft has taken into consideration in a parametric way, which starts with the minimum possible dimension of the internal components and goes up to the user defined maximum limits. The optmization was done by using *CDSGN* program

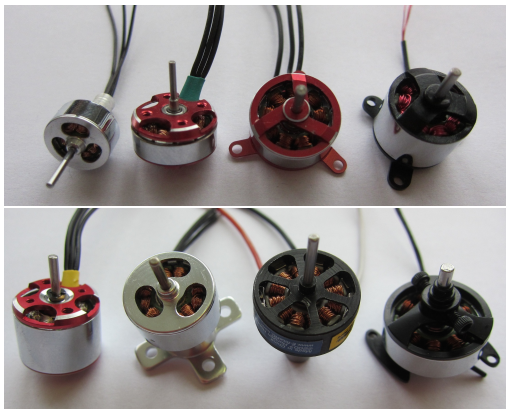


Figure 3: Pictures of candidate commercially available off-the-shelf mini and micro electric motors

Motor Name	Mass [g]	Kv [rpm/V]
1404N	9.2	2290
E-Flite Park 180	8.0	2200
ADH50XL	7.5	1700
HK AP05	5.8	3000
HexTronik 5g	5.0	2000
A05	4.3	2900
ADH30S	2.7	7500
HexTronik 2g	2.0	7700

Table 2: Coefficients of candidate commercially available off-the-shelf mini and micro electric motors

http://www.imavs.org/

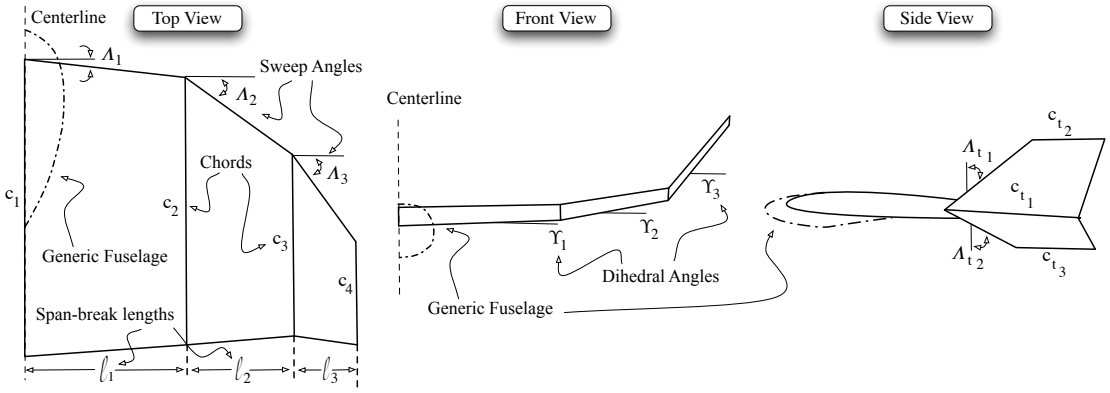


Figure 4: Main parameters for the wing planform that were variable during the multi-disciplinary optimization of the **QUARK** MAV by *CDSGN* program

through changing the wing planform (plus the vertical component of the wing), battery capacity, and cruise speed for candidate motor and propeller combinations. Figure 4 shows the main parameters that were variable for the wing planform while optimizing performance of the QUARK MAV.

5 System Integration and Design

It is crucial to take into account the system integration during the design phase. Finally, the limitation on the battery shape, servo choice and GPS module (because of their availability and ordering issues by a research lab) restricted the minimum dimensions of the parameters and created a solid core system that has to be fitted into the vehicle. According to the center of gravity and the wing thickness issues, the placement of the components have been fixed without too much option. Figure 9 shows the internal placement of the components in the final design. Note that the requirement of placing every component as close as possible to the leading edge comes from the center of gravity alignment. Figure 5 shows the first prototype that is build from EPP foam by using a CNC hot-wire cutting machine and the custom designed smallest Paparazzi autopilot for the **QUARK** MAV.

6 Sensor modelling

All the sensors embedded in the Quark autopilot are low-cost and so have imperfection. The major error sources in the navigation system are due to : -the inertial instrument imperfections (the sensor itself); -the numerical integration process in the INS mechanization. Inertial sensors have both deterministic and random errors. Deterministic errors are due to manufacturing (characteristized by the manufactuers and generally over-estimated with respect to effective values [1]) and mounting defects. This can be obtained using calibration procedure. On the other hand, the stochastic errors are random errors that occur due to random variations of bias or scale factor over time. The accurate modeling of the MEMS sensor errors is one of the most challenging tasks in the design of low-cost navigation. Indeed, to overcome the disadvantages associated with autonomous operation of the GPS and INS, the two systems are often paired together and the INS/GPS data fusion is commonly permormed using Kalman Filter (KF)[5]. There are more recent derivatives of KF known as GMEKF[8] and π-IUKF

http://www.imavs.org/



Figure 5: The prototype **QUARK** MAV wit the new developed **QUARK** Paparazzi Autopilot (October 2012)

http://www.imavs.org/

[7]. KF has widely been used for data fusion but there are, however some inadequacies, including the following :

- the requirement for a priori information of the system and measurement covariance matrices, R and Q, for each sensor;
- the necessity to have a gaussian assumption on probability distribution and a white bandwith noise.

A general model for the sensor output can be described by the following equation [9] :

$$\zeta_m = (1 + S_f)\zeta_t + b(t) + n(t) \tag{1}$$

where ζ_m is the mesured quality at the sensor output and ζ_t is the true quatity. S_f represent scale factor error (In this work, we will ignore the scale factor error $S_f = 0$ i.e, calibrated out at the factory), $b(t)$ represents the time varying bias (random drift) and $n(t)$ is observation noise, which is assumed to be band limited white noise with spectral density R. This can be characterized by taking the standard deviation of the sensor's output over a short period of time with no input applied. The bias term $b(t)$ is a random process consisting of the integral of a white noise process $\omega(t)$ with spectral density Q. The bias signal is modeled as follows :

$$\dot{b}(t) = \omega_b, \quad E\{\omega(t_1)\omega(t_2)\} = Q\delta(t_1 - t_2) \tag{2}$$

	Static Bias	σ	$R = \sqrt{\sigma^2 \cdot dt}$	R (datasheet)	Q
X-acceleration	0.142	0.0319	1000	400	$9.09e^{-5}$
Y-acceleration	-0.3	0.0985	3100	400	$7.26e^{-5}$
Z-acceleration	0.19	0.049	1600	400	$4.02e^{-4}$
Roll	-1.55	0.0825	0.0082	0.0224	$3.6e^{-3}$
Pitch	-1.13	0.1673	0.016	0.0224	19 e^{-3}
Yaw	-1.7	0.2214	0.047	0.0224	1.9 e^{-3}

Table 3: Covariance and static error for INS sensors

GPS	σ	
X	0.88	m
Y	1.45	m
Z	20.29	m
V_X	0.248	m/s
V_Y	0.146	m/s
V_Z	0.280	m/s

Table 4: GPS variance

In order to extract the covariance and static bias of each sensor noise afecting the ouput, we have analyzed 1000 samples of each sensor using a PDF (Probability Density Function). These data

coming from hight frequency dataloger system working at 100 Hz (dt=0.01s) for INS sensor and 5 Hz (dt=0.2s) for GPS. Remark that, a simplest Allan variance analysis[6] with a record of 2 hours of data has been conducted to obtain spectral density Q. The determined models for each of the sensors (gyrometers and accelerometers) are tabulated in Table 3, and shows that spectral density values R (μg or $^\circ/s/\sqrt{(Hz)}$) given by the manufacturer are generally overestimated or underestimated with respect to effective values. For example, a spectral density value of $0.0082^\circ/s/\sqrt{(Hz)} = (0.0082 \cdot 60)^\circ/\sqrt{(hr)}$ on the roll axis rotation rate indicates, that the angular error (incertainty) due to random walk is e.g. 0.063 deg after 1minute or 3.8 deg after 1hour. The first two column represents static bias and standard deviation (σ) and the last column Q represents spectral density of $\omega(t)$ coming from Allan variance analysis. In terms of visual comparison, only gyrometers histogram are shown in Fig.N, and match with a gaussian distribution. Note that it is the same shape for accelerometers histogram. Finally, GPS data variance are tabulated in Table 4 for speed and position.

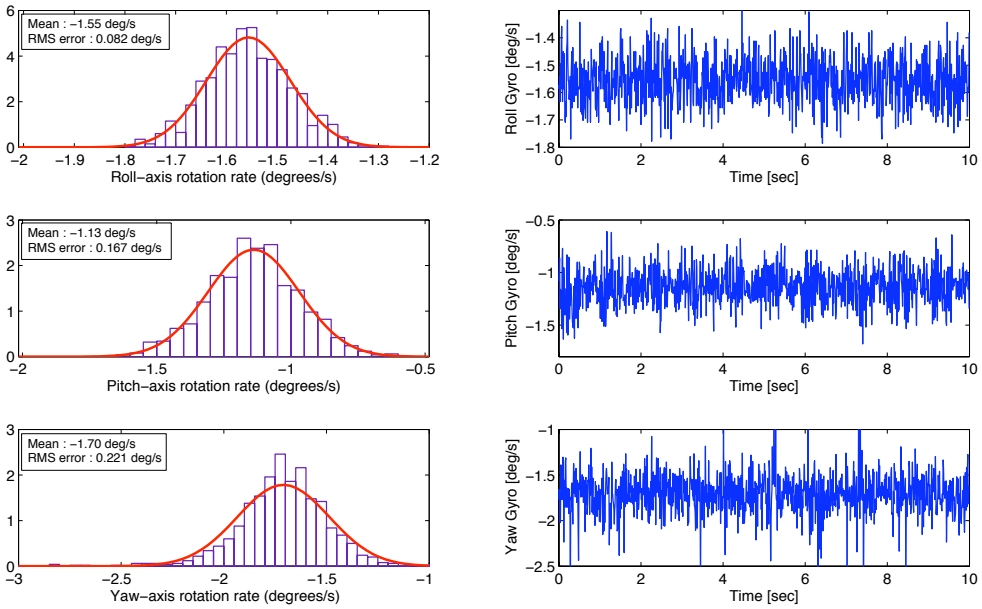


Figure 6: Distribution and RMS error of the used gyrometer

Inertial sensor always contain a certain amount of measurement noise (e.g, vibration). On the sensor level, this noise is usually separated into long-term (low-frequency) and short-term (high-frequency) noise. Wavelet denoising is frequently used to process inertial sensors measurements to improve the signal-to-noise ratio (SNR). However, in the case of **QUARK** Paparazzi autopilot, it will not be possible to use wavelet denoising because it is computationally costly. Frequency Fourier Transformation (FFT) of both gyrometer and accelerometer, recorded with an on-board datalogging system, for main flight phases (waiting, takeoff, cruise an landing) are represented in Figure 7. FFT which depends on the throttle, shows similar characteristics on both sensors after 20 Hz which can be considered as vibration noise. Thus, a Low Pass Filter (LPF) with a cut frequency at 22 Hz can be used to remove high frequency vibration. This LPF has been implemented to the autopilot.

http://www.imavs.org/

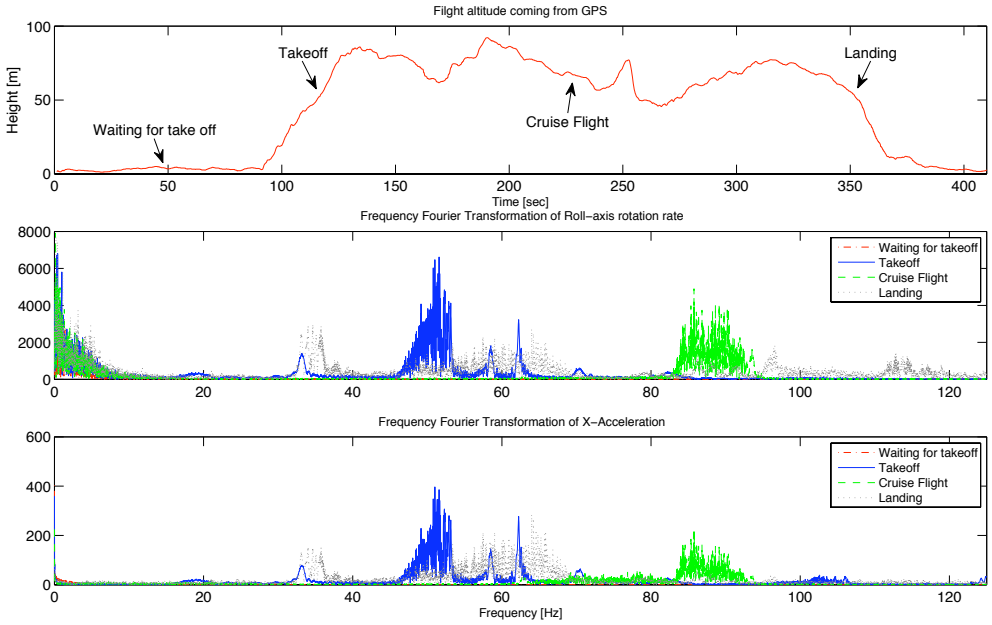


Figure 7: Frequency Fourier Transformation of roll-axis gyrometer and X-axis accelerometer values with the flight altitude information on the top.

7 Flight Tests

7.1 Longitudinal Tuning

The difficulty of flying such a small aircraft is obvious, so that prior to the outdoor flight tests, it is decided to tune the longitudinal control loop gains with a restricted flight method, that is shown in Figure 8, and tune the longitudinal control loop gains prior to the maiden flight.

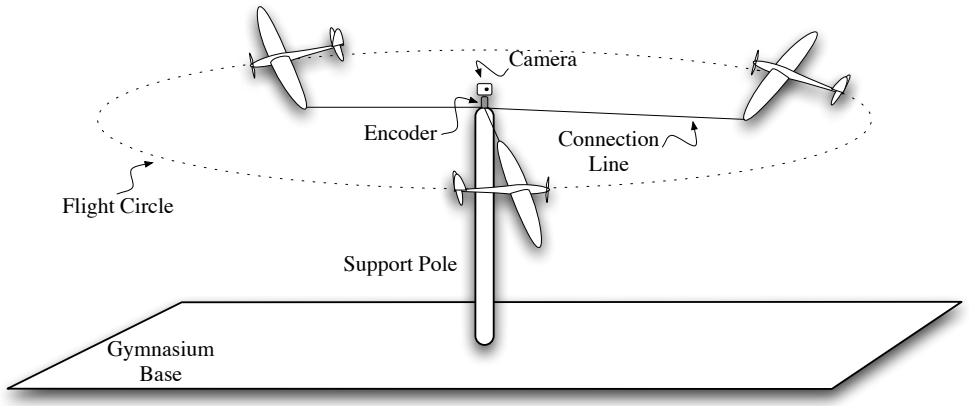


Figure 8: The longitudinal control test setup for restricted flight tests

http://www.imavs.org/

The experiment is done in a closed place, where a pole is used as the center fixation point. The aircraft is connected from its wing-tip to an encoder which is placed on top of the fixation pole. The encoder was turning freely and measuring the exact rotation rate during the experiment. Additionally, a high frame rate camera is placed on top of the encoder directed to the aircraft which will be recording the vertical movement of the aircraft during the flight with a fixed reference.

The length of the line has been increased gradually. The center of gravity was located at aircraft's aerodynamic center because of component integration issues resulting a highly agile vehicle. Therefore, the experiment started directly with autopilot assisted mode where the pilot controls the attitude of the vehicle through autopilot (fly-by-wire). The longitudinal control loop gains tuned while watching the oscillation of the vehicle and the line length has been increased up to the wall limits.

As a result of the experiment, several important flight parameters are obtained such as, the longitudinal control loop gains, the minimum flight speed, approximate cruise throttle setting, flight angle of attack at different speeds. Additionally, the control surface travels obtained for the pitch control have been copied for the roll control travels as a starting point for the outdoor flight.

http://www.imavs.org/

7.2 Outdoor Flights

The first parameters and gains that are obtained from the pole flight tests are applied for the maiden flight. The flight has been started directly in autopilot assisted mode as explained in the previous section. After a short attempt, a small angle of attack increment was required in order to sustain the level flight. The pitch proportional gains has been increased (10%) in comparison to the values that are obtained from the pole flight tests. However the lateral behavior of the vehicle was satisfactory directly from the beginning (in no-wind condition). Although the maiden flight took less then 5 minutes, the amount of battery energy used for the flight gives an estimated 20 minutes of flight time for the current prototype with small batteries.

The second flight has been made in a windy day with more than $10\,m/s$ wind gusts. The main objective of this flight was to tune the navigation control loop gains in order to fly completely autonomously. Unfortunately, $20\,m$ after the launch the telemetry link started to degrade and as the safety pilot radio command link was transmitted through the telemetry, it was no more possible to take back the flight control of the vehicle. However, the starting navigation control loops were sufficient enough to keep the vehicle in the air following the predefined flight circle on top of experiment area. Therefore, we had the chance to change some of the stabilisation and navigation control gains whenever we catch telemetry connection.We have observed severe rolls because of the strong wind gusts several times, and the counter force applied in order to correct them was seemed to be not sufficient and fast enough. That leads us to increase the lateral gains which should theoretically improve the response for the wind-gusts, unfortunately before being able to change and tune the gains (lack of constant communication), the vehicle crashed because of a strong wind gust. Thanks to its light weight and low inertia, beside a broken propeller there was no other major damage.

The third flight has been made in calm weather conditions with some modifications to the telemetry link with the hopes of improved the communication. Unfortunately, the telemetry behavior was the same as the previous flight limiting the further tuning of the vehicle. As the weather was calm with less wind, the vehicle flew much better and follow the desired trajectory closely compared to the previous flight. After 5 minutes of autonomous flight, the plane started to loose altitude at a point of the trajectory where the communication was lost. It finally crashed in a field where it was not possible to retrieve it. Without any telemetry data, the reason for this crash remain unknown.

Despite having lost the only prototype, we have obtained precious information from the flights. The prototype showed the feasibility of flying smaller than $20\,cm$ vehicle with cheap off-the-shelf

components. It was also a good opportunity to verify and correct our aerodynamic codes for this scale which gives a big confidence for the future designs.

8 On-going Work

According to the experience gained from the prototype design and flights, a new version of the **QUARK** MAV is designed with 15.9 cm wing span. This time the manufacturing of the vehicle is going to be done by using precisely machined molds by CNC. The total take-off weight is 67 g. Figure 9 shows the CAD model of the new **QUARK** with the already manufactured aluminium molds.

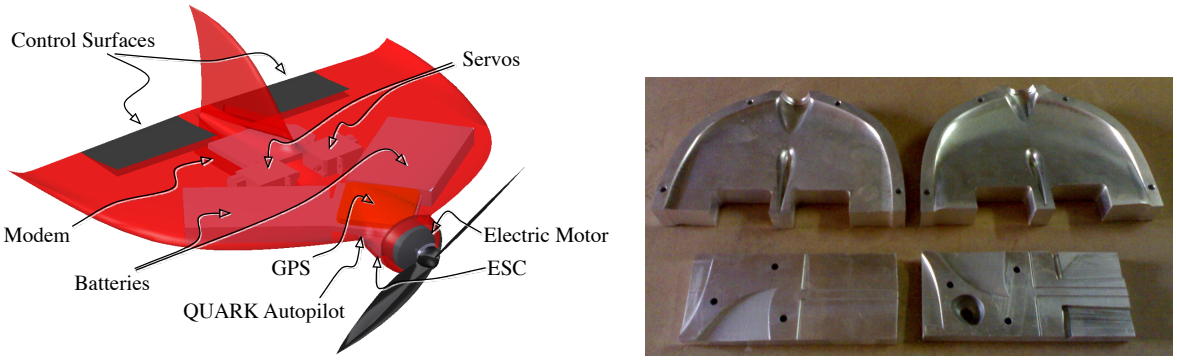


Figure 9: The CAD representation of the final **QUARK** MAV with the internal components and the aluminium CNC molds on the right

As a continuation and future work of this study, the telemetry integration will be taken into consideration more seriously in order to improve the communication. Additionally in order to make this scale of MAVs useful for real-life applications such as surveillance and information gathering from an hazardous place, a micro camera and its transmitter will be integrated into system. The camera integration has been already included on the final **QUARK** design.

9 Conclusion

Present work explains the efforts given for the realisation of an MAV with a maximum size of smaller than 20 cm. The first prototype (18 cm) is designed and manufactured by using off-the-shelf components. Complete autonomous flight including stabilisation and navigation has been achieved with the prototype. The results obtained from the limited number of flights shows a good confidence for the final design (15.9 cm) to fly over 35 minutes at a flight speed of 15 m/s and reach up to 30 km for the straight line range at a speed of 16 m/s.

Acknowledgements

The authors would like to thank to Matthieu Navarro, Michel Gorraz for their effort on developing the smallest Paparazzi Autopilot and Alexandre Bustico for helping on the sensor measurements.

http://www.imavs.org/

References

[1] Mattia De Agostino, Ambrogi Maria Manzino, and Marco Piras. Performance Comparison of Different MEMS-based IMUs. In *Position Location and Navigation Symposium (PLANS) 2013 IEEE/ON*, may 2010.

[2] Murat Bronz. *A Contribution to the Design of Long-Endurance Mini Aerial Vehicles*. PhD thesis, l'Ecole National de l'Aviation Civile, 2012.

[3] Mark Drela. An analysis and design system for low reynolds number airfoils. In University of Notre Dame, editor, *Conference on Low Reynolds Number Airfoil Aerodynamics*, June 1989.

[4] Mark Drela. *QPROP Formulation*. MIT Aero and Astro, June 2006.

[5] Kalman R. E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[6] El-Sheimy N., Hou H., and Niu X. Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on Instrumentation and Measurement*, 57(1):140–149, 2008.

[7] Condomines J. P., Seren C., and Hattenberger G. Nonlinear state estimation using an invariant unscented kalman filter. In *AIAA Guidance Navigation and Control Conference*, pages 1–15, Boston, MA, August 2013.

[8] Martin P. and Salaün E. Generalized multiplicative extended kalman filter for aided attitude and heading reference system. In *AIAA Guidance Navigation and Control Conference*, pages 1–13, August 2010.

[9] L. Wenger and D. Gebre Egziabher. Systems Concepts and Performances Analysis of Multi-Sensor Navigation System for UAV Applications. In *2nd AIAA Unmanned Unlimited Systems Conference, San Diego, CA, AIAA*, 2003.

http://www.imavs.org/

Hexrotor UAV Platform Enabling Dextrous Aerial Mobile Manipulation

Guangying Jiang[1] and Richard Voyles[2]

[1] University of Denver, Denver, Colorado, USA
gjiang2@du.edu
[2] Purdue University, West Lafayette, Indiana, USA
ricard.voyles@du.edu

Abstract

Mobile manipulation is a hot area of study in robotics as it unites the two classes of robots: locomotors and manipulators. An emerging niche in the field of mobile manipulation is aerial mobile manipulation. Although there has been a fair amount of study of free-flying satellites with graspers, the more recent trend has been to outfit UAVs with graspers to assist various manipulation tasks. While this recent work has yielded impressive results, it is hampered by a lack of appropriate testbeds for aerial mobile manipulation, similar to the state of ground-based mobile manipulation a decade ago. The Collaborative Mechatronics Lab is addressing this instrumentation gap with the development of a dexterous UAV platform to eventually host a low-cost, lightweight Stewart-Gough platform that can be combined as a macro/micro mobile manipulation system. Based on the concept of force closure (a term from the dexterous manipulation community), the new type of dexterous, 6-DoF UAV provides the unique capability of being able to resist any applied wrench, or generalized force-torque. Typical helicopters or quadrotors cannot instantaneously resist or apply an arbitrary force in the plane perpendicular to the rotor axis, which makes them inadequate for complex mobile manipulation tasks. We have developed a hexrotor UAV that can exert arbitrary wrenches in the 6-DoF force/torque space. In this paper, we describe the importance of force closure for mobile manipulation, explain why it is lacking in current UAV platforms, and describe how our hexrotor provides this important capability as well as exhibiting holonomic behavior. We also describe the results of a staged peg-in-hole task that exerts forces without pitching and rolling the UAV, reducing uncertainties.

http://www.imavs.org/

1 Introduction

The field of mobile manipulation combines two broad classes of robots, locomotors and manipulators, yet the potential impact is greater than the sum of the parts. Aerial mobile manipulation is an emerging field within mobile manipulation for which the locomotor is a UAV (unmanned aerial vehicle) [11, 15, 1, 9]. The popular quadrotor has become the main UAV of choice in robotics research, due to its ease of control and low cost. The added mobility and access that quadrotors provide brings a new dimension to the study of mobile manipulation and new challenges, as well.

One of the greatest challenges that UAVs, in general, introduce, and quadrotors in particular, is the inability to instantaneously exert forces in the plane. Quadrotors are non-holonomic; in order for them to move forward or sideways, they first have to pitch the entire body of the quadrotor to direct the thrust vector in the desired direction. What this means for aerial mobile manipulation is that the quadrotor cannot resist an arbitrary generalized force/torque. In the parlance of the dexterous manipulation community, it lacks "force closure". In fact, in one of the first attempts to use a UAV to interact physically with its environment, Albers et

al had to add an auxiliary actuator to maintain contact so Newton' Third Law of equal and opposite reaction would not immediately push the UAV away [2].

To solve this specific problem more elegantly, we are developing a hexrotor UAV that can instantaneously resist arbitrary forces – in other words, it provides force closure. To perform precise and effective mobile manipulation, this is a property that any locomotor must have, be it ground-based, water-based, or air-based. To achieve this, the thrusters of our hexrotor are canted so that the combined thrust vectors span the space of Cartesian forces and torques. This adds little cost or complexity to a conventional hexrotor.

The focus of this paper is on the hexrotor platform and its suitability as a future carrier of a manipulator for aerial mobile manipulation. Although we do describe the preliminary design of a low-cost, lightweight hexmanipulator (6-DoF Stewart-Gough platform) we have in development, that is not the focus of this paper.

1.1 Force and Form Closure

Force closure, as defined by Rimon and Burdick, is the ability of a mechanism to directly resist any arbitrary wrench (combination of force and torque) applied to it. A "force closure grasp" in the dextrous grasping literature [13, 3], is a grasp of an object that can resist an arbitrary wrench applied to the object. This class of grasps is important to the dexterous manipulation community but is often ignored by the mobile manipulation community because of the large mass of the mobile base and other issues have greater priority.

However, aerial mobile manipulation is a newly emerging field and the floating nature of aerial platforms and the highly compliant nature of their positioning brings the issue of force closure to the fore. Force closure for arbitrary rigid objects in 3-space requires six controllable degrees of freedom in force-torque space to truly accomplish. Most piloted helicopters and unmanned aerial vehicles manipulate objects by hanging them from a line and/or employ fewer than six vehicles [12], severely limiting the range of wrenches that can be applied to the object. The quadrotor teams of Mellinger et al [11] rigidly clamp to the manipulated object but, due to their design, they can only apply limited forces and torques. In generalized force space, they are effectively degenerate.

Furthermore, it is not sufficient to simply attach a 6-DoF manipulator to the bottom of a quadrotor or other degenerate aerial platform, as this does not guarantee force closure. While a 6-DoF manipulator can exert arbitrary wrenches when grounded, if the base upon which it is mounted cannot resist an arbitrary wrench, the combination remains degenerate.

1.2 Related Works

People have started to apply helicopters and quadrotors to mobile manipulation. Unstable dynamics of the vehicle and coupled object-aircraft motion while grasping objects during flight has been studied[15]. For the manipulation system as UAVs equipped with a gripper, contact forces and external disturbances acting on the gripper and the entire system should be considered[16, 1, 4]. And there have also been works using multiple collaborative UAVs in order to perform transportation tasks[7, 6]. However, when it comes to multiple UAVs carrying one payload, there are problems like the interactions between UAVs, physical couplings in the joint transportation of the payload and stabilizing the payload along three-dimensional trajectories. In these studies, single or a team of helicopters or quadrotors with grippers have been used to assist various manipulation tasks and yielded impressive results. But when trying to manipulate an object, these UAVs cannot exert arbitrary forces possibly applied to the object.

http://www.imavs.org/

Because typical helicopters or quadrotors cannot instantaneously resist or apply an arbitrary force in the plane perpendicular to the rotor axis.

Typical quadrotors have four fixed pitch propellers with parallel (and vertical) thrust vectors. The four thrust actuators provide a mapping from 4-D actuator space to 6-D Cartesian force space with rank no greater than four. In fact, the standard quadrotor configuration results in rank of exactly four as the four thrusters provide linear force along Z, and torques around X, Y and Z. Torque around Z is achieved indirectly through coriolis forces resulting from differential angular velocities of the counter-rotating propellors. (This is what makes tri-rotors infeasible and conventional single-rotor helicopters also have four actuators: main rotor, tail rotor and two actuators on the swash plate.) Instantaneous exertion of linear forces along X and Y are impossible with these configurations (as are instantaneous velocities in X and Y, resulting in a nonholonomy).

Typical helicopters and quadrotors are inherently underactuated for their 6-DoF mobility of position and orientation in space. With only four independently controlled inputs, they cannot independently control UAV's position and orientation at the same time. To fix this problem, a quadrotor design with tilting propellers has been presented in [10]. With four additional actuators added for tilting propellers, the 'quad-tilt UAV' now has full control over its 6-DoF mobility. When the propellers tilt, all the thrusters become nonparallel. And developed about the same time as our design, a similar non-planar design called 'Tumbleweed' was introduced by the University of Manchester [5]. Using six variable-pitch/fixed-speed rotors, it is designed to truly achieve full flight envelope, as hover at any orientations and translate in any directions. In these two works, the actuation concept of tilting propellers during flight actually makes it possible to access all 6 degrees of freedom of the robot. But for aerial manipulation tasks, we want to exert forces as fast as possible to resist any arbitrary wrench. Tilting propellers during flight using servos may not be fast enough for our purposes. So each rotor on our hexrotor is pitched at an optimized angle based on specific task.

2 Design

To create six independent degrees of freedom in force/torque space, one must have at least six actuators. A conventional hexrotor has six parallel-thrust propellers spaced evenly around the circumference of a circle. Because all thrusters are vertical, no components of the six thrust vectors point along the X or Y axes; they all point out-of-plane, resulting in no in-plane components. So we rotate each thruster a cant angle ϕ around its radius to form a nonparallel design, in-plane components result while still maintaining a symmetric basis of vectors.

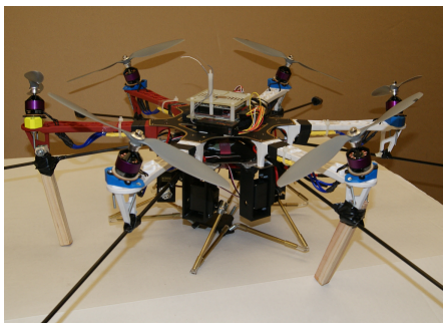


Figure 1: Flight-capable prototype of Dexterous Hexrotor with manipulator.

http://www.imavs.org/

The motors are mounted on in house designed and fabricated ABS plastic adapters that canted at an optimized angle tangentially to the edge on the end of each arm. Position of six motors and their rotation are defined in Fig. 2. By alternating clockwise and counter-clockwise rotations, the torque each motor produced shares same direction with motor force's in-plane components, providing torque around Z axis. X configuration is chosen, so X axis aligns with $Motor_3$ and $Motor_6$.

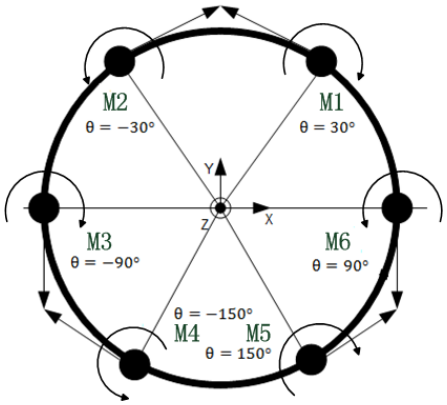


Figure 2: Motor definitions of Dexterous Hexrotor with rotation indicated, where n is the motor number, θ represents angular displacements of the motors.

2.1 Force Decomposition

The thrust and torque applied on the UAV by each motor can be expressed as

$$\begin{aligned} F_{motor} &= K_1 * PWM_{motor} \\ \tau_{motor} &= K_2 * PWM_{motor} \end{aligned} \tag{1}$$

where K_1 and K_2 are motor-dependent parameters and can be determined experimentally. PWM_{motor} is the command torque sent to the motor via Pulse Width Modulation (PWM).

To compute the net force/torque acting on the UAV from all thrusters, we first decompose each motor's thrust and torque into X, Y, and Z components on the body frame. The components of Cartesian generalized forces from $Motor_1$ can be put into a matrix as in equations (2) and (3).

$$\begin{bmatrix} F_{1fx} \\ F_{1fy} \\ F_{1fz} \\ F_{1\tau x} \\ F_{1\tau y} \\ F_{1\tau z} \end{bmatrix} = \begin{bmatrix} -K_1 * PWM_1 * cos(\theta_1) * sin(\phi) \\ K_1 * PWM_1 * sin(\theta_1) * sin(\phi) \\ K_1 * PWM_1 * cos(\phi) \\ d * K_1 * PWM_1 * cos(\theta_1) * cos(\phi) \\ -d * K_1 * PWM_1 * cos(\theta_1) * cos(\phi) \\ d * K_1 * PWM_1 * sin(\phi) \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} \tau_{1\tau x} \\ \tau_{1\tau y} \\ \tau_{1\tau z} \end{bmatrix} = \begin{bmatrix} -K_2 * PWM_1 * cos(\theta_1) sin(\phi) \\ K_2 * PWM_1 * cos(\theta_1) sin(\phi) \\ K_2 * PWM_1 * cos(\phi) \end{bmatrix} \tag{3}$$

http://www.imavs.org/

Where $[F_{1fx}F_{1fy}F_{1fz}F_{1\tau x}F_{1\tau y}F_{1\tau z}]^T$ are forces and torques decomposed from the thrust produced by $Motor_1$ and $[\tau_{1\tau x}\tau_{1\tau y}\tau_{1\tau z}]^T$ are Coriolis effects resulting from the torque produced by $Motor_1$. ϕ is the cant angle of each thruster and note that when ϕ is zero, the Coriolis effect only produces a yaw torque as with a normal quadrotor. θ represents the motor's position relative to the body and d is the radius of the UAV.

Then we compute the total force/torque $[F_{1x}F_{1y}F_{1z}\tau_{1x}\tau_{1y}\tau_{1z}]^T$ applied on the UAV by $Motor_1$ as

$$\begin{bmatrix} F_{1x} \\ F_{1y} \\ F_{1z} \\ \tau_{1x} \\ \tau_{1y} \\ \tau_{1z} \end{bmatrix} = \begin{bmatrix} F_{1fx} \\ F_{1fy} \\ F_{1fz} \\ F_{1\tau x} + \tau_{1\tau x} \\ F_{1\tau y} + \tau_{1\tau y} \\ F_{1\tau y} + \tau_{1\tau z} \end{bmatrix} = PWM_1 * \begin{bmatrix} -K_1 C\theta_1 S\phi \\ K_1 S\theta_1 S\phi \\ K_1 C\phi \\ C\theta_1(dK_1 C\phi - K_2 S\phi) \\ C\theta_1(-dK_1 C\phi + K_2 S\phi) \\ dK_1 S\phi + K_2 C\phi \end{bmatrix} \tag{4}$$

Once one motor is decomposed, we can follow the same pattern and decompose all six motors. Then we can combine these six 6x1 matrices into a 6x6 matrix which is the thrust to force/torque mapping, M_ϕ. With K_1 and K_2 determined for our motors, if the cant angle $\phi = 20°$, the thrust mapping is $M_{20°}$

$$M_{20°} = \begin{bmatrix} -1.69 & 1.69 & 0 & -1.69 & 1.69 & 0 \\ 0.97 & 0.97 & -1.95 & 0.97 & 0.97 & -1.95 \\ 5.36 & 5.36 & 5.36 & 5.36 & 5.36 & 5.36 \\ 1.21 & 1.21 & 0 & -1.21 & -1.21 & 0 \\ -0.7 & 0.7 & 1.4 & 0.7 & -0.7 & -1.4 \\ 0.65 & -0.65 & 0.65 & -0.65 & 0.65 & -0.65 \end{bmatrix} \tag{5}$$

which would conform to a nonparallel design. This matrix provides a mapping from 6-D actuator space to 6-D force/torque space and has a rank of 6, indicating we have 6 independent controlled degrees of freedom in Cartesian force/torque space.

To control the force/torque applied to the platform, the desired Cartesian force/torque vector is multiplied by the inverted thrust mapping matrix, producing a vector of PWM values (motor command torques) as in equation (6):

$$\begin{bmatrix} PWM_1 \\ PWM_2 \\ PWM_3 \\ PWM_4 \\ PWM_5 \\ PWM_6 \end{bmatrix} = [M]^{-1} \cdot \begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \tag{6}$$

2.2 Control

A control system based on equation 6 for indoor human controlled flight has been developed as shown in Fig. 3. An attitude controller is used to stabilize the hexrotor for hunman controlled flight and control hexrotor's orientation. An position controller is implemented to control hexrotor's position. Hexrotor's attitude is measured by a 9 Dof IMU and its position in space is measured by an 8-camera Vicon MXT40 motion capture system.

Unlike common quadrotor, orientation and position control is separate in hexrotor's control system. τ_x, τ_y, τ_z are used to keep and adjust hexrotor's orientation. $[F_x F_y F_z]$ are used to

http://www.imavs.org/

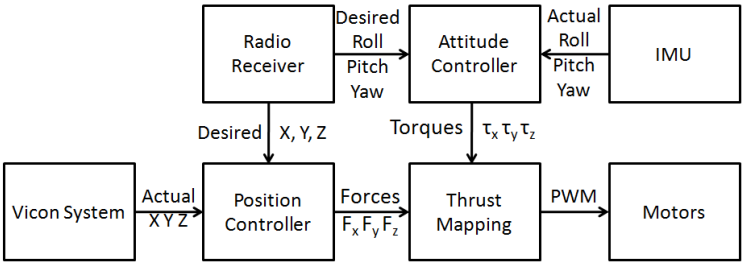


Figure 3: Hexrotor control system.

control hexrotor's thruster vector and maintain its position. SO when hexrotor's orientation is changed and tilting at an angle, its thruster vector will still point upward with adjustment of $[F_x F_y F_z]$ rather than point forward like a common quadrotor.

2.3 Optimizing the Cant Angle, ϕ

The cant angle ϕ decides how much force we can put into F_z, τ_x, τ_y or F_x, F_y, τ_z. Dependent on the motors we chosen, desired load of our manipulator and diameter of the UAV frame, we plot maximum forces and torques near hover condition at different cant angles in Fig. 4

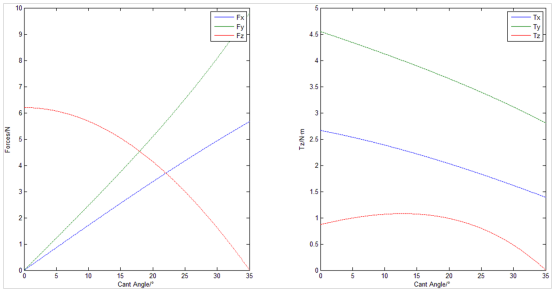


Figure 4: Maximum forces and torques near hover condition at different cant angles.

Clearly at $\phi = 0°$, there are no forces in X or Y and not much torques around Z. The opposite happens when ϕ is 35°, where all forces in Z are used to provide lift, but more force can be applied in X and Y.

To optimize the cant angle for the performance of our UAV, we adapt Yoshikawa's concept of "manipulability" to ours. As defined by Yoshikawa, "manipulability measure" is a quantitative measure of manipulating ability of robot arms in positioning and orienting the end-effectors, by looking at the isotropism of manipulator's motion in linear dimensions X, Y, Z and angles roll, pitch, yaw [17]. So for our UAV, we consider the combination of forces and torques as a similar measure of mobility. We are going to look at the isotropism of the forces and torques, not just how strong is it. To visualize how isotropic are the forces and torques, we plot these force/torque ellipsoids as shown in Fig.5.

When the cant angle is 0°, we get no ability to control F_x, F_y, but a lot of ability to control F_z. We also have ability to control τ_x, τ_y, τ_z, but not too much control over yaw because coriolis effect is weak. Then we cant the motor a little bit, we get a little bit control over F_x and F_y,

http://www.imavs.org/

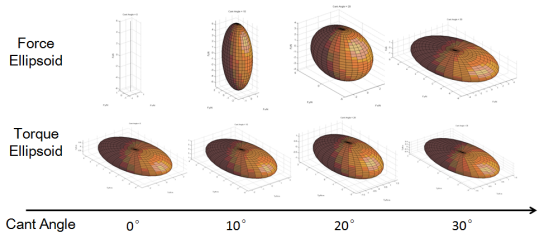


Figure 5: Force and Torque Ellipsoids at different cant angles.

but still very strong control in F_z. And likewise we get a little more control over τ_z. At 20°, the force ellipsoid becomes almost equal and the torque ellipsoid also gets more round. At 30°, the force ellipsoid gets flat and the torque ellipsoid starts to squash down again. Therefore we can see from 0° to 30° and beyond, the force/torque ellipsoid can get very isotropic at some point, which is good in our mobility measure.

We can also look at the condition number of the conversion matrix, which is the ratio of maximum eigenvalue to the minimum eigenvalue of the matrix. At 0°, when we have no control over F_x and F_y, the condition number becomes infinite, because two of the eigenvalues are 0. Then it can get smaller and smaller by increasing the cant angle, but eventually it will get higher again. Compared to a larger condition number, a small condition number means the same change in PWM would cause a smaller change in forces and torques, which is better to keep the UAV stable.

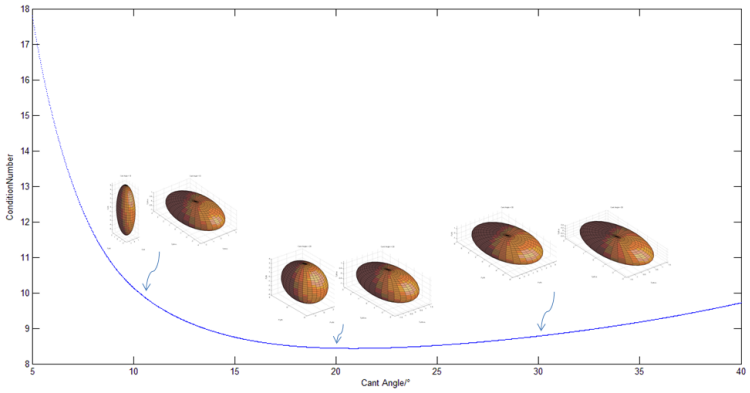


Figure 6: Condition number with force/torque ellipsoids.

We combine these two metrics on the same plot as shown in Fig. 6. We plot the condition number of the conversion matrix with force/torque ellipsoids at different cant angles, giving us a measure of the isotropism of our UAV. Therefore, dependent on the motors we have chosen, particular load of our manipulator, and diameter of the UAV frame, we optimize the cant angle at 20°. If we change the load, we would optimize this for a different cant angle.

http://www.imavs.org/

2.4 Aerial Manipulator

To complete an aerial mobile manipulation system with the aerial mobile base described above, we have begun development of a lightweight, parallel HexManipulator. This 6-DoF end effector, which augments the 6-DoF of the hexrotor, will provide a macro/micro combination for enhanced performance. The HexManipulator will provide the fine adjustments at higher bandwidth that the coarse hexrotor is unable to do. We are developing the software infrastructure on our RecoNode CPU subsystem to control the UAV/manipulator combination in a macro/micro configuration using Khatibs Operational Space formulation [8]. This manipulator platform is similar to the configuration of Uchiyama's HEXA parallel robot[14], which provides a large workspace (for a parallel-chain manipulator) based on rotary actuators.

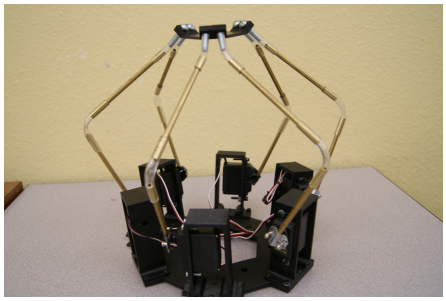


Figure 7: Hexrotor flying with manipulator.

There are six links, actuated by a rotary actuator, connecting the base and traveling plate. Each link is a serial combination of a 1 DoF active rotary joint, a 2 DoFs passive universal joint, and a 3 DoFs passive spherical joint. The link is composed of two rods connected by the universal joint and is connected to the base by the rotary joint while it is connected to the traveling plate by the spherical joint on the other end. The 2 DoFs universal joint is implemented with a length of flexible tubing and the 3 DoFs spherical joint is implemented with the same length of tubing plus a passive rotary joint that allows the tubing to twist freely.

3 Experiment

To test the response time of the Dexterous Hexrotor corresponding to external forces, a staged peg-in-hole task is presented with the hexrotor and a typical quadrotor from DIY Drones. A frame diagram as Fig. 8 shows the experiment's setup. Active control of the HexManipulator was not used in this test as the peg was held rigidly by the hexrotor. With the peg trapped half-way in a hole which is attached to a 6 DoFs force sensor, we start the UAV up and increase throttle until it can carry its own weight. Then we command both the hexrotor and quadrotor to exert a horizontal force perpendicular to the axis of the peg.

The result is shown in Fig. 9. Changes in force sensor measurements of Fy are direct measurements of the force which UAVs applied on the hole. A positive and a negative pulse are detected around 600ms and 1600ms. At the meantime, attitude of the UAVs was recorded and plot in the same figure.

The result shows that when both the hexrotor and quadrotor are able to exert same force in the plane perpendicular to the rotor axis, the hexrotor is much faster. This is because the hexrotor is exerting the force in a different way than common quadrotor. While quadrotor is

http://www.imavs.org/

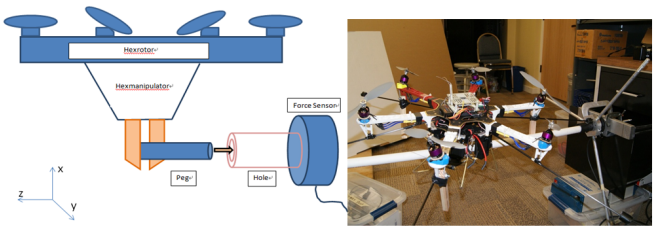


Figure 8: Peg-in-hole setup, with force sensor's coordinate system indicated.

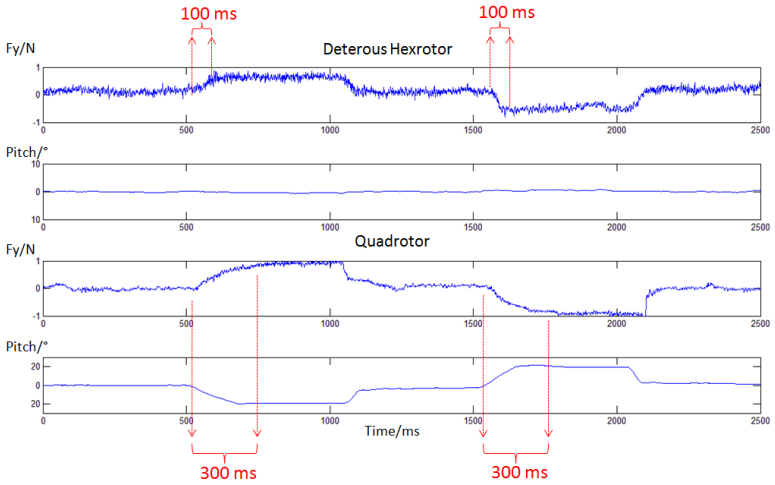


Figure 9: Hexrotor and quadrotor peg-in-hole measurements of F_y, Pitch.

tilting an angle to generate the horizontal force, the hexrotor can do it by simply changing the rotational velocity of the rotors without tilting. It is obvious that the lag varying speeds of propellers of the hexrotor will be much smaller that that caused by tilting the entire body of a quadrotor.

4 Conclusion

A truly holonomic aerial rotorcraft that provides force closure for controlled interaction with external structures has been introduced. The key contribution of this paper is to derive the thrust mapping based on decomposed net force/torque and develop a metric for the cant angle optimization of the dexterous UAV's performance for manipulation-based tasks. A flight-capable prototype has been built and tested.

It should be noted that we claim our hexrotor can "instantaneously resist arbitrary forces." This is not strictly true as the hexrotor can only change the torque of its motors instantaneously. The required change in the thrust magnitude is not dependent on torque, for a propeller, but on speed. Therefore, the independent thrust magnitudes and the resulting net force/torque experience a lag due to the inertia of the thrusters. The lag due to the rotor inertia is much smaller than that due to pitching the entire vehicle, as for a conventional quadrotor, and we believe is smaller than the pitching of the variable ϕ concept of the Tumbleweed (which also

http://www.imavs.org/

has to overcome the gyroscopic action of the rotors).

References

[1] S. Stramigioli A. Keemink, M. Fumagalli and R. Carloni. Mechanical design of a manipulation system for unmanned aerial vehicles. In *Proceedings, IEEE International Conference on Robotics and Automation*, 2012.

[2] A. Albers, S. Trautmann, T. Howard, T. Nguyen, M. Frietsch, and C. Sauter. Semi-autonomous flying robot for physical interaction with environment. In *Proceedings, IEEE International Conference on Robotics Automation and Mechatronics*, pages 441–446, June 2010.

[3] Antonio Bicchi. On the closure properties of robotic grasping. *International Journal of Robotics Research*, 14:319–334, August 1995.

[4] T. Danko C. Korpela and P. Oh. Mm-uav: Mobile manipulating unmanned aerial vehicle. In *Journal of Intelligent and Robotic Systems*, 2011.

[5] A. Scillitoe I. Lunnon A. Llopis-Pascual J. Zamecnik s. Proctor M. Rodriguez-Frias M. Turner A. Lanzon W. Crowther D. Langkamp, G. Roberts. Modeling and control of a quadrotor uav with tilting propellers. In *Proceedings, International Micro Air Vehicle conference*, 2011.

[6] N. Michael V. Kumar D. Mellinger, M. Shomin. Cooperative grasping and transport using multiple quadrotors. In *Distributed Autonomous Robotic Systems*, volume 83, pages 545–558, 2013.

[7] M. Bernard A. Ollero I. Maza, K. Kondak. Mechanical design of a manipulation system for unmanned aerial vehicles. In *Journal of Intelligent and Robotic Systems*, volume 57, pages 417–449, Jan 2010.

[8] O. Khatib. A unified framework of motion and force control of robot manipulators: The operational space formulation. In *IEEE Journal of Robotics and Automation*, volume RA-3, pages 43–53, 1987.

[9] C. Korpela and P.Y. Oh. Designing a mobile manipulator using an unmanned aerial vehicle. In *IEEE International Conference on Technologies for Practical Robot Applications*, April 2011.

[10] P. R. Giordano M. Ryll, H. H. Bulthoff. Modeling and control of a quadrotor uav with tilting propellers. In *Proceedings, IEEE International Conference on Robotics and Automation*, 2012.

[11] D. Mellinger, M. Shomin, N. Michael, and V. Kumar. Cooperative Grasping and Transport using Multiple Quadrotors. In *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*, Nov 2010.

[12] Nathan Michael, Soonkyum Kim, Jonathan Fink, and Vijay Kumar. Kinematics and statics of cooperative multi-robot aerial manipulation with cables. In *ASME International Design Engineering Technical Conference and Computers and Information in Engineering Conference*, 2009.

[13] V.-D. Nguyen. Constructing force-closure grasps. *International Journal of Robotics Research*, 7:3–16, June 1988.

[14] M. Uchiyama P. Maurine, D.M. Liu. Self calibration of a new hexa parallel robot. In *4th Japan-France Congress and 2nd Asia-Europe Congress on Mechatronics*, volume 1, pages 290–295, 1998.

[15] D. Bersak P. Pounds and A. Dollar. Grasping from the air: Hovering capture and load stability. In *Proceedings, IEEE International Conference on Robotics and Automation*, 2011.

[16] F. Ruggiero V. Lippiello. Mechanical design of a manipulation system for unmanned aerial vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[17] T. Yoshikawa. Manipulability of robotic mechanisms. In *Journal of Robotics Research*, volume 4, 1985.

http://www.imavs.org/

Technical Session 1

Rotors and propellers

Chairman : Emmanuel Benard

- Sergey Serokhvostov. ***One idea of propeller for low Reynolds numbers***
- Zhen Liu, Longlei Dong, Jean-Marc Moschetta and Jianping Zhao. ***Nano Rotor Blade Airfoil Optimization***
- Matthew Anderson, Kc Wong and Patrick Hendrick. ***Propeller Location Optimisation for Annular Wing Design***

http://www.imavs.org/

One Idea of Propeller for Low Reynolds Numbers

Sergey V. Serokhvostov

Moscow Institute of Physics and Technology
Department of Aeromechanics and Flight Engineering
140180, Gagarina street, 16, Zhukovsky, Russia
serokhvostov@phystech.edu

Abstract

The main idea is to make larger the propeller relative chord length (with respect to diameter) with the dimensions decrease. So, at several dimension this propeller will look like the marine screw. The analysis of the small marine screw design applicability for MAVs is discussed.

http://www.imavs.org/

1 Introduction

One of the difficulties for MAV dimensions miniaturization is propeller dimensions. One can see that the propeller diameter with respect to wing span of MAVs becomes greater and grater with the wing span decrease (see fig. 1). So, for some wing span the propeller diameter can be equal to the wing span if the existing technology of propeller design is used. Of cause, this situation is unacceptable if we have the limitations on the maximal aircraft size. From this, some other ways for propeller design of very small MAVs must be found. One of the proposals is to use "marine screw"-like propellers.

2 Problem statement

Very useful for the propeller performance description are dimensionless characteristics: thrust coefficient C_T, power coefficient C_P and efficiency η as function of advanced ratio J:

$$J = \frac{V}{nd}, \quad C_T = \frac{T}{\rho n^2 d^4}, \quad C_P = \frac{P}{\rho n^3 d^5}, \quad \eta = \frac{C_T J}{C_P},$$

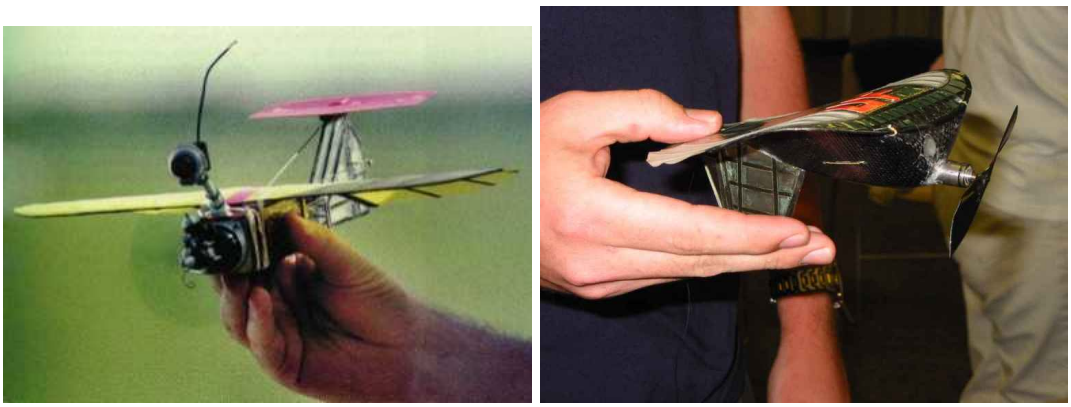


Figure 1: University of Florida MAVs

where V — air velocity at infinity, n — frequency of propeller rotation, ρ — air density, d — propeller diameter, T — propeller thrust, P — propeller power.

It is well known that for the fixed Reynolds number Re these coefficients are independent of dimensions and velocities. The behavior of these characteristics at low Reynolds numbers was investigated, for example, in [6].

Imagine that one have the propeller tuned for some flight regime (for example, the best efficiency for defined propeller diameter, flight velocity and thrust) and want to use this propeller dimensionless geometry for the other propeller diameter and flight conditions.

If one simply change the propeller size proportionally in all dimensions, then, for the same advanced ratio the thrust changes as forth power of diameter and the power changes as fifth power of diameter. But the same advanced ratio at changing the propeller diameter assume either that flight velocity is changed or frequency of propeller rotation is changed.

In the case of constant velocity the frequency must be changed inversely proportional to the diameter, so the thrust is totally changed as second power of propeller size. (If we resize the airplane in such way, the drag, lift and power required also change as second power). So, in this case some possibility of thrust and drag equivalence exists. Here we not take into account the effects of low Reynolds numbers.

But in the case of resizing dimensions and decreasing the flight velocity or in other cases one can't obtain the equivalence of thrust and drag at fixed propeller advanced ratio. So, in this case one must change either the advanced ratio or propeller geometry.

First of all, one can change the angles of profile installation along the blade and doesn't change the chords length. This way enable to optimize the propeller's characteristics in rather close limits.

Second, one can increase the number of blades. In this case the propeller efficiency can slightly decrease.

Third way is the increasing the chords length. From aerodynamical point of view for the small Re one blade is better than two blades of the same total chord. (There even exist the examples of propeller with one blade, see fig. 2.) In MAV practice from time to time one can see that the aircrafts has propeller with cut blades, see fig. 3. This method increases relative chord of propeller blade.

Increasing the chord of blade one make the propeller more and more like the marine screw. And at some moment can appear the idea of using the marine screw geometry for the MAV propeller.

http://www.imavs.org/

3 Analysis

Let's make some comparisons. As an example of "marine screw" can be taken the screws investigated in [4]. A set of screws was tested in water tunnel (see fig. 4).

Characteristic data for these experiments were: frequencies of rotation - 1700–5000 RPM, water flow velocity — 9–14 ft/s, screw diameters — 2–2.5 in, Pitch/Diameter — 1.4–2.0.

So, maximal Reynolds number (for characteristic chord length of 2 sm and velocity at the end of the blade) is about $Re = 10^5$. One must keep in mind that in main part of data points from [4] Re is lower than 10^5.

Maximal value of efficiency obtained in these experiments is about 60–70% (see fig. 5).

Also one can see from fig. 2 that the maximal thrust coefficient C_T (denoted on the graph as KT) corresponding to $J = 0$ is about $C_T = 0.5$. For the model airplane 2-blade propeller this value is about $C_T = 0.1$ [1].

It should be mentioned that the propellers analyzed were for hobby boats, i.e. hardly have the best performance available. So, they should be compared with the propellers of the "same nature" i.e. small propellers for hobby aircrafts. For example, propeller U80 has maximal efficiency of about



Figure 2: Examples of one blade propellers

40–50% for $Re = 50000$ [2]. Data from [1] also show that efficiency of small propellers is low enough, see fig. 6.

Also one should compare the above data with the propeller performance of BlackWidow MAV [3] (see fig. 7). The propeller data are: diameter — 3.81 inch, pitch — 6.04 inch ($P/D = 1.58$), frequency 5000 RPM, velocity blade tip $V_{\text{tip}} = 25$ m/s, $Re = 50000$, efficiency — 80%.

4 Discussion

At the times of "aviation childhood" and "aviation youth" there was the division of air propellers and marine screws, see. fig. 8 [5]. As we can see above, with the decreasing the dimensions of MAV it is better to make the propeller blade wider and angle of blade installation bigger. This enables to obtain more thrust at nearly the same or better efficiency.

Also it should be noted that the propellers for hobby boats and aircrafts are not with the best efficiency, and example of Black Widow propeller shows that the proper design can improve the performance.

http://www.imavs.org/



Figure 3: MAV propeller with cut blades



Figure 4: Screws tested in [4]

The reason of usage of "aircraft propellers" can be explained by the fact that for propeller efficiencies of 90% and more the part of efficiency losses due to the thrust is comparable with losses due to friction drag and it is well known that larger diameter for the same thrust gives better efficiency.

But in the case of restrictions on aircraft dimensions (and especially for MAVs) the part of efficiency losses due to friction becomes much more larger that for higher efficiencies. In this case it's better to use the propellers with wider blades. So, an some size the propeller can looks like "marine screw" (one can say that even a Black Widow propeller looks a little like marine screw).

The design methodology for marine screw is well developed for boats and other water vehicles. Moreover, a lot of good screw designs were made. For a set of MAV applications it can be preferable to use the existing marine designs.

http://www.imavs.org/

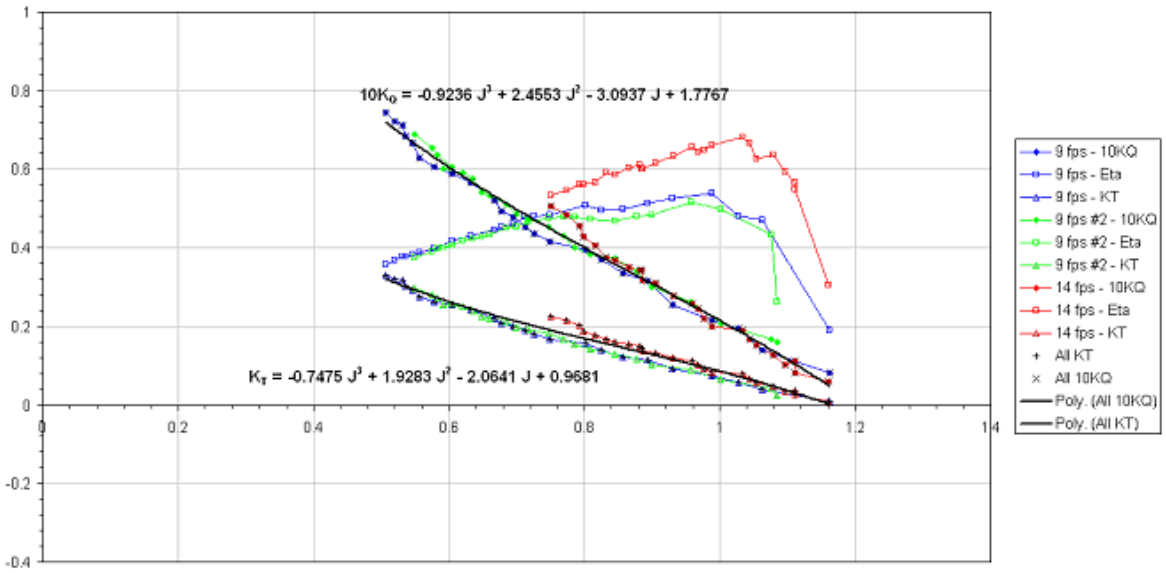


Figure 5: The corrected propeller performance curves for propeller 1457 [4]

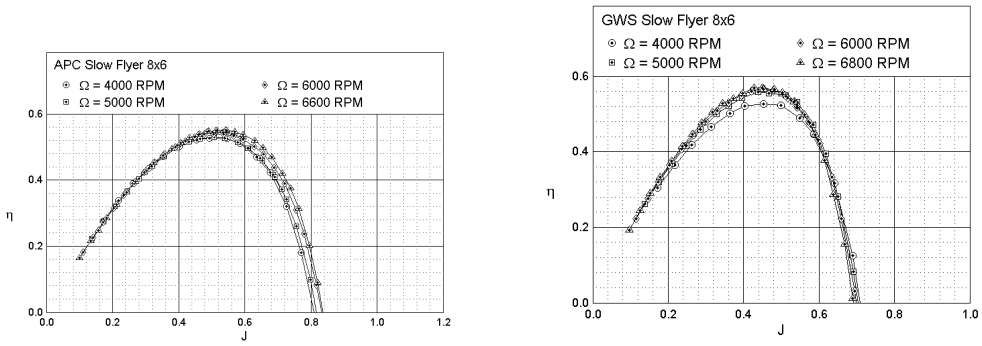


Figure 6: Examples of propeller efficiency, Re =40000–70000 [2]

5 Conclusion

The main conclusion from the above analysis is that it can be preferable to use the propellers with wide blades at higher angles of installation in MAV designs. It can provide higher thrust at nearly the same efficiency. Very good initial point for such propellers design can be the shapes of marine screw.

http://www.imavs.org/

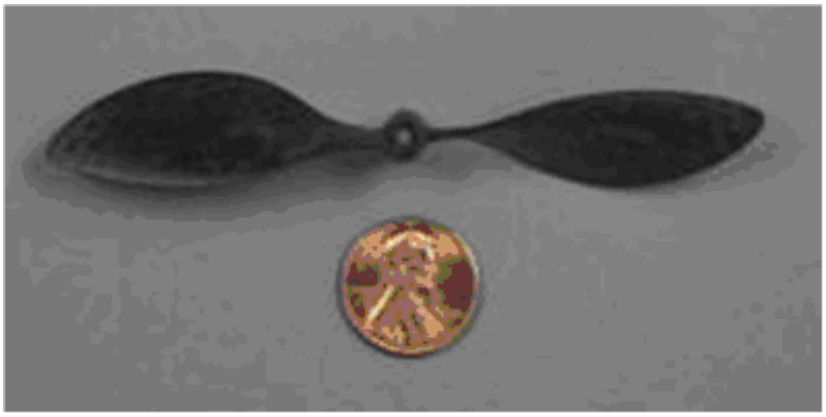


Figure 7: Black Widow MAV propeller

http://www.imavs.org/

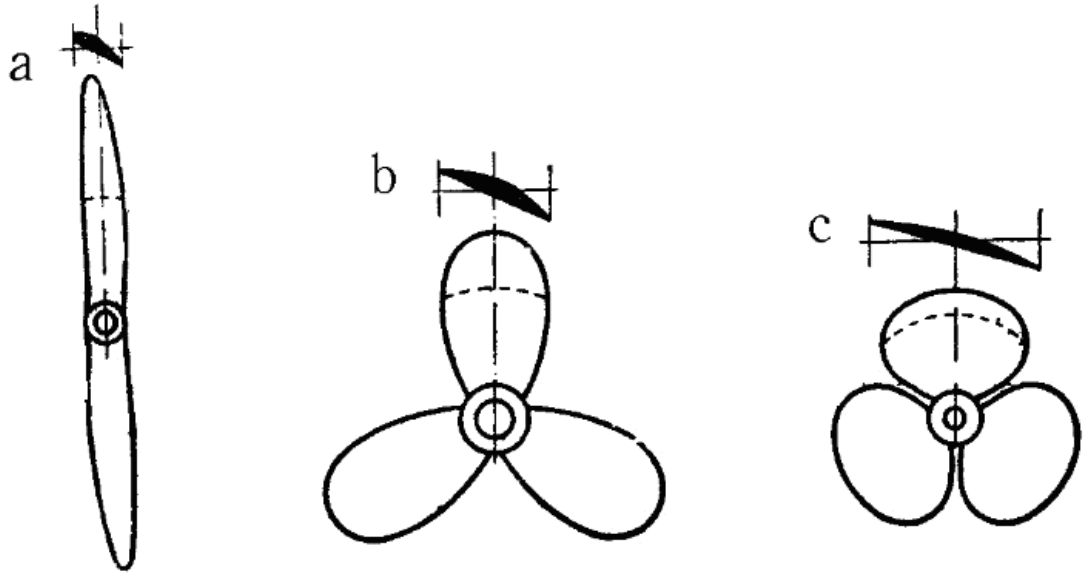


Figure 8:

References

[1] Uiuc propeller database. [online]. Available at `http://www.ae.illinois.edu/m-selig/props/propDB.html`.

[2] Francis Barnhart, Michael Cuipa, Daniel Stefanik, and Zachary Swick. Micro-aerial vehicle design with low reynolds number airfoils. [online], 2004. Brigham Yound University. Available at `http://rajaero.yolasite.com/resources/MAV_Micro%20Air%20Vehicle%20Design%20Project%20-I.pdf`.

[3] J. M. Grasmeyer and M. T. Keennon. Development of the black widow micro air vehicle. *AIAA Paper*, 2001.

[4] Jessica A. Jacobson. Using single propeller performance data to predict counter-rotating propeller performance for a high speed autonomous underwater vehicle. Master's thesis, Virginia Polytechnic Institute and State University, 2007.

[5] L. Prandtl. *Führer durch die Strömungslehre.* Vieweg und Sohn, Braunschweig, 1944.

[6] Sergey V. Serokhvostov and Tatyana E. Churkina. One useful propeller mathematical model for MAV. In *Proc. IMAV*, 2011.

http://www.imavs.org/

Nano Rotor Blade Airfoil Optimization

Zhen Liu[1]*, Longlei Dong[1]†, Jean-marc Moschetta[2]‡
and Jianping Zhao[1]§

[1] Xi'an Jiaotong University, Xi'an, China
liuz@mail.xjtu.edu.cn,dongll@mail.xjtu.edu.cn,zhaojianping@mail.xjtu.edu.cn
[2] Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse, France
jean-marc.moschetta@isae.fr

Abstract

The aerodynamic performance of airfoil at ultra-low Re has a great impact on the propulsive performance of nano rotor. Therefore, the optimization of airfoil is necessary before the design of nano rotor. Nano rotor blade airfoil optimization is a multi-objective problem since the airfoil suffers a wide range of Reynolds number which increases the difficulty of optimization. In this paper,the airfoil of nano rotor was optimized based on the controlled elitist Non-dominated Sorting Genetic Algorithm II (NSGA-II) coupling with the parameterization method of Class function/Shape function Transformation technique (CST) and the multi-objectives function processing method of statistic definition of stability. An airfoil was achieved with a thickness of 2% and a maximum camber of 5.6% at 2/3 of chord. Optimized airfoil exhibits a good aerodynamic performance at ultra-low Re according to the simulation with CFD solver based on 2D incompressible NS equation. And comparisons were carried out between the performance of the rotors designed with designed airfoil and ag38 airfoil, which showed that the optimized airfoil was suitable for rotor design.

http://www.imavs.org/

Contents

1 Introduction **2**

2 Geometry Parametric Representation and Aerodynamic Solver **4**
2.1 Geometry Parametric Representation 4
2.2 Aerodynamic Solver 5

3 Objective Function Definition Method and Optimization Method **7**
3.1 Objective Function Definition Method 7
3.2 Optimization Method 7

4 Results and Discussion **9**

5 Conclusion **13**

6 Future Work **14**

7 Acknowledgments **14**

*Lecturer, State Key Laboratory for Strength and Vibration of Mechanical Structures, school of Aerospace; corresponding author.
†Associate Professor, State Key Laboratory for Strength and Vibration of Mechanical Structures, school of Aerospace.
‡Professor, Department of Aerodynamics, Energetics and Propulsion.
§Engineer, State Key Laboratory for Strength and Vibration of Mechanical Structures, school of Aerospace.

http://www.imavs.org/

1 Introduction

Rotary-wing Nano Air Vehicle (NAV) is typically a configuration for NAV design. The propulsive performance of rotor determines the flight performance of NAV directly. It is therefore necessary to design the rotor so as to obtain an excellent propulsive performance. As the aerodynamic performance of nano rotor blade airfoil remarkably influences on nano rotor performance, the airfoil optimization is an important part of rotor design. However, the study of nano rotor airfoil is rarely performed. SAMARAI, a rotary-wing NAV, employs directly AG38 thick airfoil [44], and the other design of nano rotor selected the existing low-Re airfoil as well [17]. Therefore, the optimization of nano rotor airfoil requires to be carried out.

Typical airfoil optimization includes the selection of aerodynamic solver, the process of objective function, the geometry parametric representation of airfoil and airfoil optimization method etc. It is a challenge to combine all the parts together to design an excellent airfoil for nano rotor because of its special working condition. Firstly, nano rotor optimization requires an efficient and accurate aerodynamic solver at ultra-low Reynolds number. Nano rotor operates at ultra-low Reynolds number lower than 20000 because of its small dimension resulting in difficulties of optimization. Since the computational resource and time are enormous for an airfoil optimization, an accurate and efficient solver is preferable. Navier-Stokes solver coupling with grid generation method are utilized for optimization of airfoil [34, 41, 38] in recent years but it requires enormous computational cost. Vortex panel method method such as XFoil is also popular for optimization of airfoil due to its time-saving characteristics especially with fairly good results at low Reynolds number. Besides, the optimization of nano rotor blade airfoil is a multi-objective problem to achieve an airfoil with excellent aerodynamic performance along different stations of rotor blade. Because nano rotor are usually both tapered and twisted and the relative velocity of blade section is proportional to the distance from the rotational center along the blade, nano airfoil blade rotor suffers a wide Reynolds number ranging from 6000 to 16000 taking into account the hub. So the optimized airfoil shall process excellent aerodynamic performance at different Reynolds numbers. Kunz [24] utilized the lift-to-drag as the objective to optimize the airfoil separately at two Reynolds numbers. For the single-point optimization, the aerodynamic performance of airfoil may degrade at off-design point. Most optimizations take the value of objectives at design points as objective functions resulting in the irregular airfoil form especially for multi-objective optimizations. Some studies utilized multi-point optimization method to take into account several Res [11]. However, the optimal airfoil was always achieved with corrugated surface. Li and Padula [26] proposed statistical definition of stability method in which the mean value and the standard deviation of values of objectives at design points are taken as the final objective functions. This method ensures the optimized results at each design points and the smoothness of airfoil form.

Airfoil optimization method is also a key point for nano rotor blade airfoil optimization. In the past decades, many methods, which are basically divided into two categories: inverse method and direct method, have been proposed to get the form of airfoil directly with the flight condition. Selig et al. [36, 35] developed a multi-point inverse method to design airfoil with conformal mapping. With the development of computational techniques and algorithm method, the direct methods which allow the process of more complicate problem were applied on the optimization of airfoil. Kunz and Kroo [21] performed computation study of airfoil for rotary wing at Re from 2,000 to 6,000 and the optimizations of NACA airfoil based on gradient method. Nemec [28] and Burgree [6] utilized gradient-based numerical optimizations methods to fulfill the aerodynamic design problem. Alexandron et al. [2] applied the Approximation and Model Management Framework (AMMF), which has the capacity of rapid and early integration of

high fidelity nonlinear analyses and experimental results into the multidisciplinary optimization process, in the airfoil optimization. Other methods such as Nelder-Mead [7] and Differential Evolution (DE)[27] were performed in the airfoil design as well. A widely used method shall be GA method [29, 42, 45, 1, 31], since GA method has a great deal of merits. Oyama et al.[29] designed an airfoil with GA methods using two-dimensional NS solver. The maximum lift-to-drag ratio was taken as the objective and B-spline function was used to parameterize airfoil. Small surface waves were found on the designed airfoil. Jones [20] combined GA method with XFOIL to optimize an airfoil with the objectives of maximum lift-to-drag ratio and minimum acoustics. Airfoil was parameterized with B-spline function. And high irregular airfoil surface was obtained. Those methods employ either conventional GA method, or modified GA method. The diversity and fitness can't be ensured. Controlled elitist Non-dominated Sorting Genetic Algorithm II (NSGA-II) method proposed by Deb [9] which increases the diversity of population while keeps the elitists in the population is a potential candidate for airfoil optimization method.

The geometry parametric representation of airfoil influences the optimization results as well. From the results of references [29] and [20], airfoils were produced with small surface waves for multi-point optimization. It is analyzed that the application of bad parameterization method and objective function results in those bad results. The geometry representation is necessary before the optimization. Numerous methods [33, 32, 39, 30, 18, 37] including discrete airfoil coordinates method, Bezier or B-Spline control point representation method, free form airfoil representation, polynomial surface representation and cubic spline control point representation, were devised to numerically represent airfoil geometry. Good parametric method can construct the airfoil curve with fewer variables. Selection of a parameterization technique is an important step for airfoil shape design optimization. The parameterization based on the B-Spline has advantages of continuous second-order derivative, fewer design parameters to express various airfoil shapes and intuitive definition of initial design space [29]. However, B-Spline technique produces airfoils with small surface waves for multi-point optimization. Free-form parameterization method can prevent easy manipulation, but it is the lack of intuitive control and the inherent difficulties encountered when trying to generate airfoil-like shapes. Some special parameterization methods such as Hicks-Henne function [18], PARSEC [37] were also developed for certain shape optimization. Kulfan [23, 22] proposed Class function / Shape function Transformation (CST) method including class function and shape function to parameterize geometric shape. This method requires fewer variables to represent airfoil and ensures the smoothness of form.

From the analysis above, it is found that Inverse method and direct method are widely utilized to get the airfoil for a special case directly. An inverse method allows the velocity distribution to be directly controlled rather than anticipated from geometry perturbations, while a direct method allows the design of airfoil with taking into account multi disciplines and multi points. The direct method has more opportunity to get a global optimized solution. Therefore, some study [13] combines the indirect method and direct method in order to eliminate the disadvantages. However, the increase of computational resource and capacity of achieving an optimization method are still under discussed. The representation of airfoil and the process of objective functions have a great impact on the airfoil design since they even determine the efficiency and precision of optimization. The airfoil of nano rotor surfers a large scope of Re so that the airfoil design is a multi-objective optimization. The determination of optimization method coupling with an accurate and efficient aerodynamic solver, an appropriate parameterization method and the process method of objective functions is necessary to obtain a nano rotor airfoil with excellent aerodynamic performance.

In this paper, the airfoil of nano rotor was firstly parameterized with the robust Class func-

http://www.imavs.org/

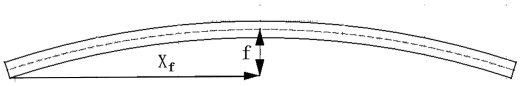


Figure 1: Basic form of plate airfoil.

tion/Shape function Transformation technique (CST). Secondly, the accuracy of aerodynamic solver was verified at ultra-low Reynolds. Then, the objective functions were processed by a method based on the statistical definition of stability. Finally, the airfoil was optimized using controlled elitist Non-dominated Sorting Genetic Algorithm II (NSGA-II) method to ensure the robustness and diversity of the population. And the aerodynamic performance of designed airfoil was computed with two-dimensional (2D) incompressible N-S solver. Comparisons were carried out between the performance of the rotors designed with designed airfoil and ag38 airfoil.

http://www.imavs.org/

2 Geometry Parametric Representation and Aerodynamic Solver

2.1 Geometry Parametric Representation

2.1.1 Basic form of blade airfoil

With the development of MAV, aerodynamic performance of airfoil at low Reynolds turns out to be an interesting field. Enormous experiments and computations have been conducted [43, 40, 16, 21, 5]. Despite the fact that the airfoil presented in those studies can't be used as the airfoil of nano rotor, they can provide this design with some guidelines. A majority of studies show that the thin plate airfoil exhibits better aerodynamic performance than thick NACA airfoil especially when the thickness is lower than 2% of chord. But conclusions vary each other for the other parameters such as camber, location of camber, leading edge angle and roughness etc. Furthermore, the limitation of traditional fabrication method confines the airfoil form. Consequently, the plate airfoil with thickness of 2% is selected as the basic form (Figure 1) to design rotor. However, the camber and location of maximum camber are still remained underdetermined. So optimization method was utilized in the following part to obtain those parameters.

2.1.2 CST Representation Method

In order to optimize airfoil, the airfoil shall be parameterized in advance. Since the plate airfoil was chosen as the basic form, the upper and the lower surfaces of airfoil are able to be represented. The selection of the mathematical representation of an airfoil which is utilized in aerodynamic design optimization has a profound impact on computational time and resources as the selection of the type of optimization algorithm. It also determines whether or not the geometries contained in the design space are smooth or irregular, or even physically realistic or acceptable. It affects the suitability of the selected optimization process. Therefore, the geometric representation technique shall include the characteristics of being capable of producing smooth and realistic shapes, good mathematically efficiency and numerically stability, flexibility, and robustness. After examination of several diverse geometry parametric representation methods, it was found the CST representation methodology illustrates a powerful capability to

represent a wide variety of 2-D and 3-D geometries encompassing a very large design space with a relatively few scalar parameters. CST method is therefore utilized as the geometry parametric representation. For the upper and the lower surfaces, CST functions are

$$Z_U\left(\frac{x}{c}\right) = C_{N_2}^{N_1}\left(\frac{x}{c}\right) \cdot \sum_{r=0}^{n} A_{U,r+1} K_{r,n} \left(\frac{x}{c}\right)^r \left(1-\frac{x}{c}\right)^{n-r} + \frac{x}{c} \cdot \frac{\triangle Z_{UTE}}{c} \tag{1}$$

and

$$Z_L\left(\frac{x}{c}\right) = C_{N_2}^{N_1}\left(\frac{x}{c}\right) \cdot \sum_{r=0}^{n} A_{L,r+1} K_{r,n} \left(\frac{x}{c}\right)^r \left(1-\frac{x}{c}\right)^{n-r} + \frac{x}{c} \cdot \frac{\triangle Z_{LTE}}{c}. \tag{2}$$

where $C_{N_2}^{N_1}$ is the class function, c is the chord length, A_r is the weight of each Bernstein polynomial item, K_r is the Binominal coefficient, and $\triangle Z_{TE}$ is the trailing edge thickness.

Plate airfoil can then be parameterized by the methodology proposed above. To ensure the smoothness of the curve, all the coefficients of Bernstein Polynomial were assumed to be positive. Since the curves of airfoil surfaces are simple, the Bernstein Polynomial is simplified to contain four items. Therefore, there are six parameters to be determined which are $X = (N_1, N_2, A_1, A_2, A_3, A_4)$. To ensue that the airfoil is physically realistic or acceptable, the bounder of array can be constrained in a small zone to reduce the computational time and resource. Since all the parameters are positive, the lower bound and the upper bound were defined to ensure that the equations can cover a wide range of airfoil shape. The results of representation method showed that the airfoil camber varied from 0% to 100% and the location of camber vary from 0% to 100% with the bound defined. And it was found that the value of exponents N_1 and N_2 defines the basic geometry of airfoil, while the value of A_1, A_2, A_3 and A_4 determine the camber and the location of maximum camber. Considering the limitation of computational solver, the leading edge and the trailing edge of airfoil were modified slightly.

2.2 Aerodynamic Solver

The optimization of airfoil requires a good optimization method as well as an accurate solver. XFOIL [12] is an analytical method whose inviscid formulation is a linear-vorticity stream function panel method. A Karman-Tsien compressibility correction with a solution generated from closely coupled viscous and inviscid methods is incorporated. It employs a two-equation, lagged dissipation, integral boundary layer solution strongly interacted with the incompressible potential flow via the surface transpiration model and an envelope e_N transition criterion which allows prediction of separation bubble. While reduced by eliminating the interactive design and plotting features, as well as modified to create a callable function, the analytic capabilities of XFOIL remain unchanged. As a result, lift, drag and moment coefficients can be obtained for airfoils operating through flight conditions in relatively brief periods of time. XFOIL was widely verified for airfoil study at Reynolds number above 15,000, but it was scarcely studied at ultra-low Reynolds. Therefore, the ability of XFOIL to predict the aerodynamic characterization of airfoil at ultra-low Reynolds were validated before the optimization. XFOIL was firstly verified with AG38 airfoil . AG38 airfoil is a thick airfoil with thickness of 7% but a maximum camber of only 2% at 0.355. Youngren et al. [44] studied the aerodynamic characterization of AG38 at low Reynolds number ranging from 15,000 to 60,000 in NASA Langley 23 Boundary Layer Tunnel. Taking account of Reynolds number at which NAV flies, AG38 airfoil was studied with XFOIL at Re of 15,000 at angles of attack ranging from $-4°$ to $8°$. Figure 2 showed that the lift coefficient and drag coefficient vary with angle of attack for both computational results and experimental results. XFOIL predicted the aerodynamic performance well at the whole range of

http://www.imavs.org/

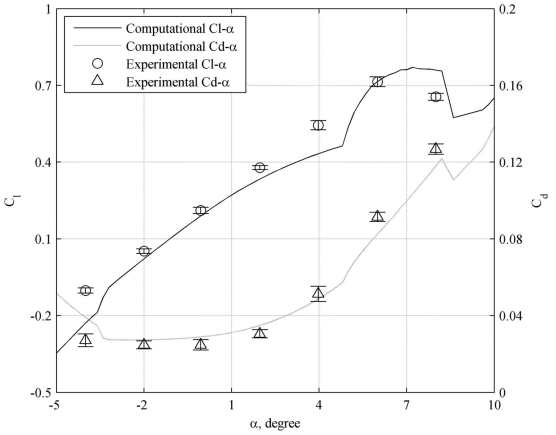


Figure 2: Comparison of computational results with experimental results [44] of AG38 airfoil at Re 15,000.

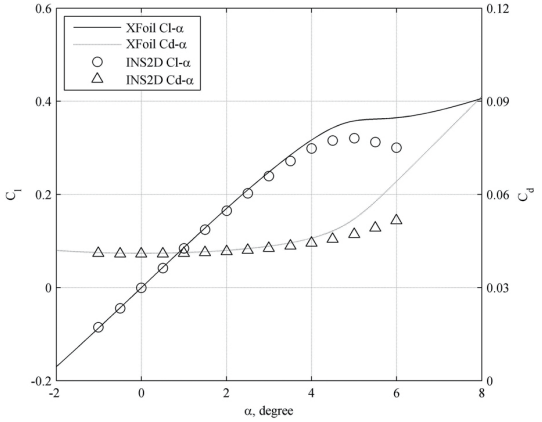


Figure 3: Comparison between computational results for XFOIL and computation results for INS2D [24] of NACA 0006 airfoil at Re 6000.

angle of attack. However, the separation of laminar was predicted earlier since the slope of lift coefficient curve drops slightly at low angles of attack. Accordingly, the value of lift coefficient computed is lower than that from experiments at angles of attack ranging from 2° to 4°, vice verse for drag coefficient. At negative angles of attack lower than −4°, the lift coefficient was under-estimated. One possible reason might be the early estimation of the separation at the upper surface of airfoil. Validation was performed with NACA0006 airfoil at Re of 6,000 as well. Kunz [24] studied the aerodynamic performance of NACA airfoil with INS2D code at low Reynolds from 2,000 to 8,000. INS2D code is two-dimensional incompressible Navier-Stokes solver developed by Rogers in which artificial compressibility method is utilized to deal with incompressible flow. This code was validated by Kunz at low Re. In this validation, NACA0006 airfoil was studied with XFOIL at Re of 6,000 to compare with Kunz's result calculated with INS2D. Figure 3 shows the comparison between the results calculated by XFOIL and Kunz's

http://www.imavs.org/

results. The lift coefficient predicted by XFOIL match well with that predicted by INS2D except at high angle of attack. Once stall appears, XFOIL fails to capture the rapid drop of the lift coefficient at angle of attack of 5°. But for the drag coefficient, XFOIL predicts higher value than INS2D after stall.

In summary, XFOIL can predict lift coefficient and drag coefficient well at Re ranging from 6,000 to 15,000 before stall. However, XFOIL predicts the separation earlier. Once attachment appears, lift coefficient is always overestimated.

3 Objective Function Definition Method and Optimization Method

3.1 Objective Function Definition Method

Prior to the procedure of optimization, the objective functions shall be determined. Through the analysis of modern MAVs, it is found that one of the bottlenecks is the power efficiency during the flight, especially with the drop of the flight speed and the decrease of flight vehicle size. Hovering performance is therefore an important parameters to judge the performance of NAV so that it shall be paid more attention during NAV design. To characterize the hovering performance, FM is defined as shown in the following equation

$$FM = \frac{C_T^{3/2}}{2C_P}. \tag{3}$$

According to the blade element theory [25], it was found that FM is a function of $C_l^{3/2}/C_d$ of the airfoil. Therefore, the objective of this optimization is to obtain the maximum value of $C_l^{3/2}/C_d$ of airfoil with the constraint of minimum lift coefficient.

The flow condition suffered by airfoil varies along the blade. Thereafter, the optimization is a multi-objective problem to achieve an airfoil with excellent aerodynamic performance along different stations of rotor blade. Several Res can be treated as the design points since Re varies along the rotor blade. Taking into account the hub of nano rotor, Re range was defined from 6000 to 16,000. Six design points were chosen to optimize the airfoil with equal interval of Re. At each design point, the angle of attack increases from $-2°$ to $10°$ to search for the maximum $C_l^{3/2}/C_d$ with the lift coefficient greater than a specified value $C_{l,spe}$.

A method based on the statistical definition of stability [26] is utilized in the present study to process objectives. The objective to obtain a maximum value of $C_l^{3/2}/C_d$ for this optimization was transformed to obtain a minimum value of $C_d/C_l^{3/2}$ to be able to apply the method. The mean and variance of $C_d/C_l^{3/2}$ with respect to Reynolds are two objective functions defined as,

$$\begin{cases} Minmize f_1(D,Re) & = & E(min(C_d(D,Re)/C_l^{3/2}(D,Re))) \\ Minmize f_2(D,Re) & = & \sigma^2(min(C_d(D,Re)/C_l^{3/2}(D,Re))) \\ subject\ to: & & C_l \geq C_{l,spe} \\ D \in X, & & Re \in (6000, 16,000) \end{cases} \tag{4}$$

3.2 Optimization Method

GA is an optimization algorithm based on Darwin's survival of the fittest evolutionary concept, according to which a biological population evolves over generations to adapt to the environment

http://www.imavs.org/

http://www.imavs.org/

by selection, recombination and mutation. It was originally described by Holland in 1960s [19]. Since then, GA was well-developed by Holland and his student, namely Goldberg. With the first application of GA to the practice by Goldberg [14], it has been increasingly applied to engineering design and optimization problems. Aerodynamic shape optimization as one of important issues of aerodynamic study profits the development of GA in recent years. The aerodynamic shape optimization of rotor airfoil is a multi-variable and multi-objective problem. GA method can solve this problem by locating a global optimum robustly. For multi-objective optimization, GA finds a local Pareto front for multiple objective functions. The fitness and diversity of population are two criterions to judge the optimal population. The controlled elitist Non-dominated Sorting Genetic Algorithm II (NSGA-II) [9] was therefore applied in the study. NSGA-II applies an elite preserving mechanism and a fast non-dominated sorting procedure assuring the preservation of good solutions found in previous generation and a fast computation. The elimination of tunable parameter increases the independence of the method to user. However, the elite solutions are composed of all solutions belonging to the currently best non-dominated front. A situation which might occur is that not enough new decision variables can be accepted in new population due to the preservation of elitism in previous generation, i.e. the diversity of population is bad. A suboptimal solution set is obtained instead of a global optimal solution. Therefore, the controlled elitist NSGA-II approach was utilized in the design.

3.2.1 NSGA-II

After the determination of objective functions, the process of GA method can be performed. At the beginning, the initial population $G_0 = (D_1^0, D_2^0, \cdots, D_j^0, \cdots, D_N^0)$ is generated randomly according to the bound of the design space. The initial population contained N chromosomes. Each chromosome is composed of six variables. The six variables named genes in the chromosome are capable of representing the geometry of the airfoil. Each gene is generated randomly with their lower and upper bounds.

The fitness functions are defined based on the objective functions as shown below.

$$\begin{cases} F_1 = f_1^{max} - f_1(D, Re) & f_1(D, Re) < f_1^{max} \\ 0 & \text{else.} \end{cases} \tag{5}$$

$$\begin{cases} F_2 = f_2^{max} - f_2(D, Re) & f_2(D, Re) < f_2^{max} \\ 0 & \text{else.} \end{cases} \tag{6}$$

In the function, f_1^{max} and f_2^{max} are the maximum values estimated for objective functions. With initial population, each chromosome is inputted into the XFOIL solver so that the objective functions are calculated and the correspondent fitness function values are achieved as a result. The fitness values for the jth chromosome are represented as $F_{1,j}$ and $F_{2,j}$. After the determination of all fitness values for each chromosome, the population is sorted based on non-domination. The fast sort algorithm proposed by Deb [10] is used in this optimization. Thereafter, all the chromosomes in the population are successively ranked with the no-dominated sort. However, there are usually several individuals in one front. In order to compare these individuals, the crowding distance function is introduced to find the Euclidian distance between each individual. All the individuals are assigned a crowding distance value. The crowing distance is then calculated based on their objectives. Thus, each individual in the population has two properties that are no-domination rank defined as P_{rank} and crowding distance Dis. The non-dominated sort has a computational complexity of $O(MN^2)$where M is the number of genes and N is the number of chromosomes.

With the results of sort, a new child population was generated with the binary tournament selection, the recombination and mutation operators. Binary tournament selection has the advantages of high efficiency, translation and scaling invariant, and easy realization of parallel evolutionary algorithms [15]. Therefore, the initial population is processed using binary tournament selection. In order to generate the offspring population of initial population, the simulated binary crossover [3, 8] is utilized to continue process the population obtained from the selection. With the population generated by simulated binary crossover, the polynomial mutation [8] is carried out to generate the final child population defined as GC^0. However, the child population GC^0 isn't the next generation parent population as the other genetic algorithm. The NSGA-II utilizes a combination of parent population G^0 and its child population GC^0 obtained in the initial step to carry out the optimization to ensure elitism as below.

$$R = G^0 \cup GC^0 \tag{7}$$

The fast non-dominated sort is carried out with R as described above. From the first front, the individuals which have high dominance are added to next generation parent population G^1 until the size exceeds the total number of chromosome N. G^1 is the next generation parent population which will be used to produce the next iteration until the convergence.

3.2.2 Controlled Elitist NSGA-II

Despite the fact that the mutation can increase the diversity of population, NSGA-II discards the chromosomes in the pareto front with high rank which deteriorates the diversity of population, especially for the case when the population is mostly comprised of currently best non-dominated solutions. Therefore, a controlled elitist non-dominated sorting GA [9] is utilized here. A geometric distribution is introduced as

$$n_i = N \cdot \frac{(1-r)r^{i-1}}{1-r^K} \tag{8}$$

where n_i , N, r, and K are the maximum number of allowed individuals in the ith pareto front, the number of individuals in the parent population, the reduction rate defined by users and the number of total pareto front, respectively. Because the next parent population has a number of individuals of N, the formula ensures the summary of n_i to be N. If n_i is higher than the number of individuals in ith pareto front, the extra number will be succeeded to the n_{i+1} until the end of sort. In the end the next generation parent population is generated with high fitness and diversity. The procedures above continued until the stopping criteria.

4 Results and Discussion

The nano-rotor blade airfoil optimization was carried out at ultra-low Re using the method above. An initial population of 100 individuals and the number of maximum generation of 200 were set for the controlled elitist NSGA-II method. In order to compute the fitness of each population, XFOIL solver was executed for one hundred times. And in each calculation, the aerodynamic forces of each airfoil shall be computed at several angles of attack. Therefore, the iterations consume huge computational resource. Hereafter, the stop criteria is defined as $f_1(D, Re) < 0.075$ and $f_2(D, Re) < 0.0075$. The stop criteria were satisfied after 120 generations. The pareto front was shown in Figure 4. The results show good diversity along the pareto front which ensures the airfoil parameters are well optimized.The airfoil optimized which

http://www.imavs.org/

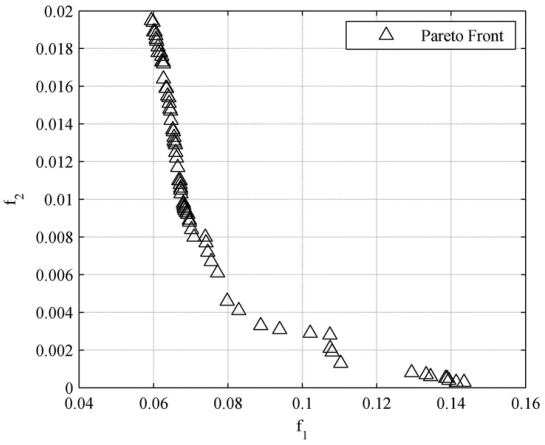


Figure 4: Pareto front of airfoil optimization.

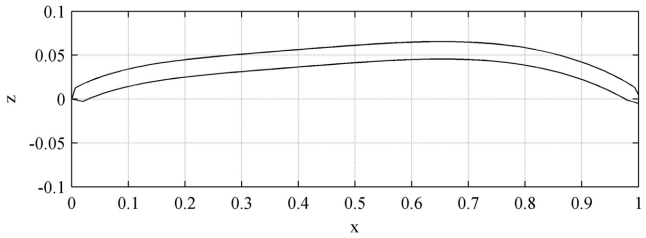


Figure 5: Geometry of airfoil optimized.

satisfies the requirement of stop criteria is illustrated in Figure 5. The airfoil has a uniform thickness of 2% with modification at leading edge and trailing edge. The maximum camber is 5.6% at the location of about 2/3. The camber of airfoil is around 5% from 0.4c to 0.8c which is different from the traditional camber line. However, the airfoil optimized approximates to that designed by Kunz for Mesicopter [24].

To verify the accurate aerodynamic performance of airfoil optimized, Computational Fluid Dynamics (CFD) method was used to simulate the flow field of airfoil. For low-Mach and low-Re flow, conventional NS equations might fail to converge to a correct solution. Therefore, 2D incompressible Navier-Stokes (INS) solver with artificial compressibility was used to compute the aerodynamic performance of airfoil optimized at ultra-low Re. Because of the special flight condition of nano rotor, the airfoil suffers flows at a wide Re range. The airfoil is simulated at typical Reynolds numbers from 6,000 to 16,000 with an interval of 2,000. At each Re, the angle of attack varies with an increment of 2°. The grid of the airfoil was shown in Figure 6. The size of mesh is 16151 points in the streamwise and the normal direction, respectively. At the leading edge and trailing edge, more grid points were distributed to capture the flow characteristics. The width of the first layer of grid is 1.0×10^{-4}.

The pressure coefficient contours at angle of attack of 4° were presented in Figure 8. According to the contours, high pressure was generated on the lower surface of airfoil while lower pressure was generated on the upper surface. And complicate vortices were found at the trailing edge of airfoil, especially at Re 16,000. So, unsteady aerodynamic phenomena appear for airfoil

http://www.imavs.org/

http://www.imavs.org/

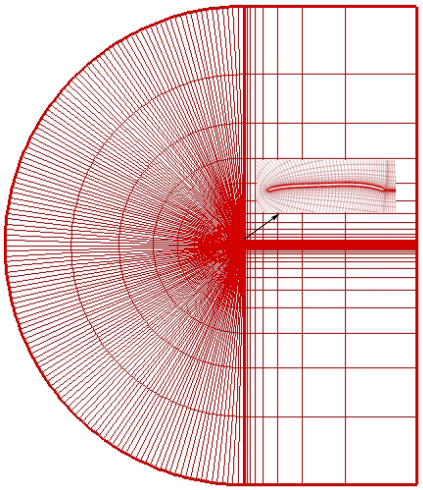


Figure 6: Grid of airfoil optimized.

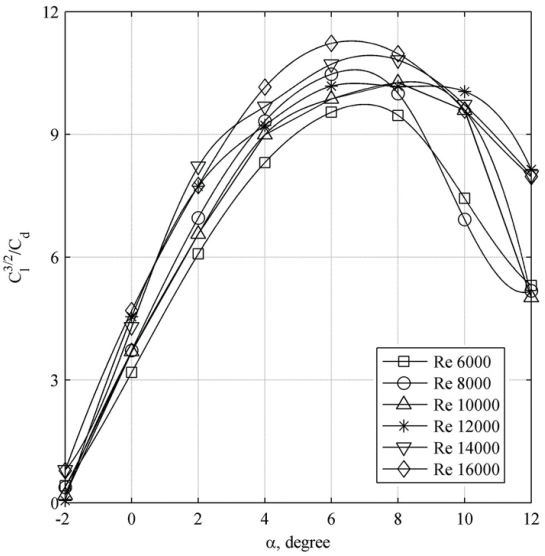


Figure 7: $C_l^{3/2}/C_d$ varying with angle of attack at different Res for optimized airfoil.

optimized at low Re. Figure 7 illustrates $C_l^{3/2}/C_d$ calculated by INS solver varies with angle of attack at Re ranging from 6,000 to 16,000 for optimized airfoil. Results show that $C_l^{3/2}/C_d$ increases with the Reynolds number in general but it varies from angle of attack as well. Generally, reaches a maximum value at angle of attack of 6°. The maximum is from 9.5 to 11.2 which are lower than that predicted by the XFOIL. However, they still reach a relatively high value at ultra-low Re when comparing with other airfoils. The maximum $C_l^{3/2}/C_d$ of airfoil AG38 is only 7.8 at Re 15,000 and 9.5 at Re 20,000 [44]. The maximum $C_l^{3/2}/C_d$ of airfoil NACA0006 at Re 6,000 is about 3.9, whereas it is about 9.5 for optimized airfoil at the same

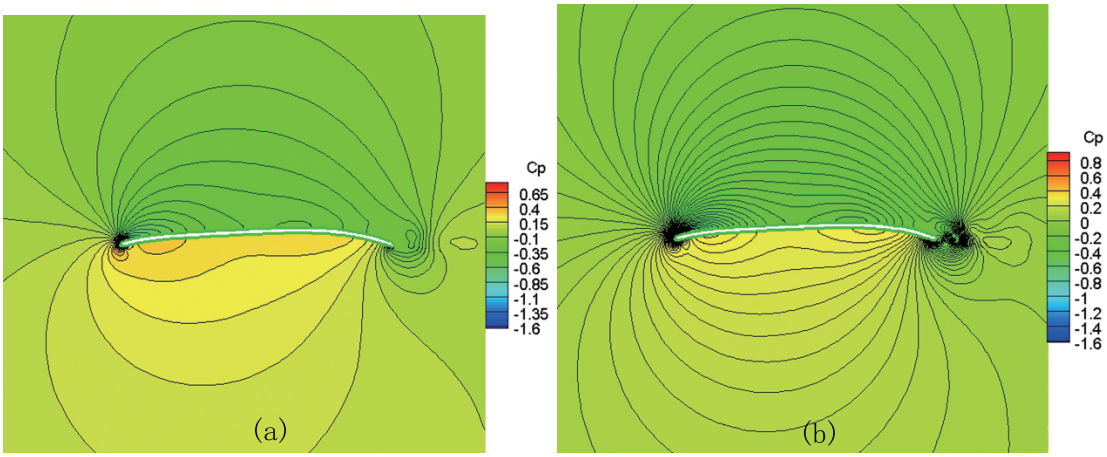


Figure 8: Pressure coefficient contours of airfoil optimized: (a)Reynolds number 8,000; (b)Rynolds number 16,000.

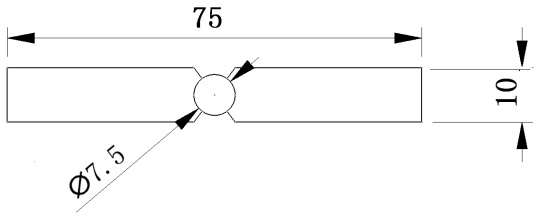


Figure 9: Simple rotor(unit:mm).

Re [24]. Lift coefficient increases while drag coefficient decreases with the increase Re, so better aerodynamic performance can be obtained at higher Re. But the irregular phenomena appear at certain Re or angle of attack due to the unsteady characteristic of ultra-low Re flow.

Simple rotors with diameter of 75mm and uniform blade chord length of 10mm were generated with optimized airfoil and ag38 airfoil as shown in Figure 9, respectively. The performance of both rotors were compared so as to verify the priority of optimized airfoil. Calculations were performed with RPM of 6500 for both rotors while pitch angle varied from 0 to 12 degree using potential goldstein formulation method. Figure 10 showed Figure of Merit(FM) varying with the ratio of thrust coefficient to solidity for both rotors. It was found that the thrust of coefficient reached a higher value for optimized airfoil. The value of FM of rotor designed with ag38 airfoil, named as rotor 2, was higher than that of rotor designed with optimized airfoil, named as rotor 2, when the thrust coefficient remained as a small value. However, FM of rotor 2 increased sharply with thrust coefficient and a maximum value of about 0.8 was obtained in comparison with only 0.5 for rotor 1. Rotor 2 exhibited a better performance than rotor 1 at high thrust coefficient.

http://www.imavs.org/

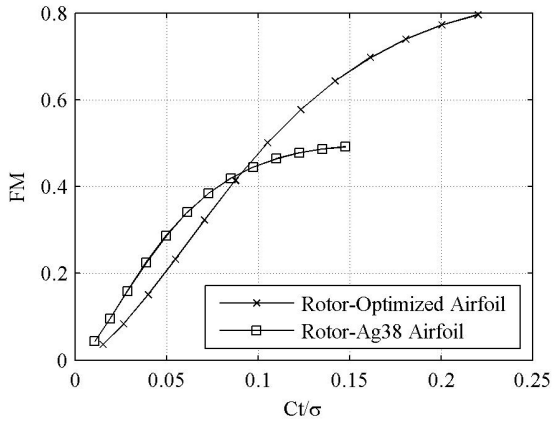


Figure 10: Figure of merit varying with ct/σ.

http://www.imavs.org/

5 Conclusion

The performance of airfoil has a direct impact on aerodynamic performance of rotor. In this paper, the optimization of airfoil for nano rotor was carried out at ultra-low Re. In order to optimize the airfoil, parameterization method which has an impact on the efficiency and precision of airfoil optimization was determined. A CST representation algorithm was selected to parameterize the airfoil after a survey of several different representation methods due to its simplicity and robustness. CST can represent current plate airfoil with a few variables and guarantee the continuity and smoothness of the airfoil form. The plate airfoil was represented with six design variables whose bounds were determined based on CST to simplify the optimization. Successively, the flow solver was verified. Because of the ultra-low Re suffered by nano rotor and the requirement of numerous computational resources, the flow solver shall have the capacities of solving the ultra-low Re flow accurately and efficiently. In this optimization, XFOIL, which is widely verified at low Re, was used. At several low Res, XFOIL solver was verified with multiple airfoils and the computational results were compared with those from experiments and CFD. It is evident that XFOIL can calculate the aerodynamic performance of different types of airfoils accurately before the stall, not only for lift, but also for drag. It is concluded that it is applicable to used XFOIL as the solver of optimization. Subsequently, the airfoil was optimized. Controlled elitist NSGA-II optimization method was employed to design nano rotor airfoil combing with the CST parameterization method and the statistical definition of stability objective-function-process method. It is innovatively proposed that the objective of this optimization is to obtain the maximum value of $C_l^{3/2}/C_d$ of airfoil instead of lift-to-drag ratio with the constraint of minimum lift coefficient. Several Res were chosen as the design points to obtain an excellent airfoil along the blade. The mean and the standard deviation of the values of objectives at the design points were taken as the objective functions based on the method of the statistical definition of stability. Controlled elitist NSGA-II method was then utilized to design nano rotor airfoil. A plate airfoil with maximum thickness of 2% and maximum camber of 5.6% at 2/3 was achieved. The optimization airfoil exhibits good aerodynamic performance at ultra-low Re. When comparing rotors' performance designed with optimized airfoil and ag38 airfoil, it was found that rotor designed with optimized airfoil can achieve a high FM and exhibited a better performance at high thrust coefficient, which proved

that the optimization methods are suitable for rotor airfoil design.

In conclusion, controlled elitist NSGA-II method combining with CST parameterization method and the statistical definition of stability objective-function-process method was successfully employed on airfoil design. A plate airfoil was designed and exhibited excellent aerodynamic performance.

6 Future Work

During the design of nano coaxial rotor airfoil, the solver of XFOIL was used. XFOIL has a certain precision to simulate airfoil at ultra-low Re, but it fails to predict the aerodynamic performance of airfoil after stall. Therefore, other solver with higher precision and efficiency can be employed. GA method has the advantages of robustness and independence from initial solutions. However, it requires a large amount of computational recourses. Therefore, GA method can be used to obtain a solution as the initial solution of the other efficient optimization method.

http://www.imavs.org/

7 Acknowledgments

The authors would like to thank the support of Natural Science Foundation of Shaanxi Province(Grant No. 2012JQ1018) and the Fundamental Research Funds for the Central Universities. They would also like to thank the laboratory of DAEP of ISAE.

References

[1] V. Ahuja and A. Hosangadi. Design optimization of complex flowfields using evolutionary algorithms and hybrid unstructured cfd. In *17th AIAA Computational Fluid Dynamics Conference*, pages 2005–4984. AIAA E-library, June 2005.

[2] N. M. Alexandron, E. J. Nielsen, R. M. Lewis, and et al. First-order model management with variable-fidelity physics applied to multi-element airfoil optimization. In *8th AIAA/USAF/-NASA/ISSMO Symposium on Multidisciplinary Analysis and optimization*, pages 2000–4886. AIAA E-library, June 2000.

[3] H. G. Beyer and K. Deb. On self-adaptive features in real-parameter evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 5(3):250–270, 2001.

[4] F. Bohorquez. *Rotor hover performance and system design of an efficient coaxial rotarywing Micro Air Vehicle*. PhD thesis, University of Maryland,US, 2007.

[5] F. Bohorquez, F. Rankins, J. D. Daeder, and et al. Hovering performance of rotor blades at low reynolds numbers for rotary wing micro air vehicles-an experimental and cfd study. In *21st AIAA Applied Aerodyanmics Conference*, pages 2003–3930. AIAA E-library, June 2003.

[6] G. W. Burgree and O. Baysal. Aerodynamic shape optimization using preconditioned conjugate gradient methods. *AIAA Journal*, 32(11):2145–2152, 1994.

[7] M. Darbandi, A. Setayeshgar, and S. Vakili. Modification of standard k-epsilon turbulence model for multi-element airfoil application using optimization technique. In *AIAA 24th Applied Aerodynamics Conference*, pages 2006–2829. AIAA E-library, June 2006.

[8] K. Deb and R. B. Agarwal. Simulated binary crossover for continuous search space. *Complex System*, 9(1):115–148, 1995.

[9] K. Deb and T. Goel. Controlled elitist non-dominated sorting genetic algorithms for better convergence. Technical report, Springer-Verlag, Heidelberg, 2001.

[10] K. Deb, A. Pratap, S. Agarwal, and et al. A fast elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transaction on Evolutionary Computation*, 6(2):182–197, 2002.

[11] M. Drela. Pros and cons of airfoil optimization. In *Frontiers of Computational Fluid Dynamics*, page 1998. World Scientific, June 1998.

[12] M. Drela and H. Youngren. Xfoil 6.94 user guide. [online], 2001. `http://www.mit.edu`.

[13] B. A. Gardner and M. S. Selig. Airfoil design using genetic algorithm and an inverse method. In *41st Aerospace Sciences Meeting and Exhibit*, pages 2003–43. AIAA E-library, June 2003.

[14] D. E. Goldberg. Genetic algorithms in search, optimization and machine learning. Master's thesis, Addison Wesley Reading, USA, 1989.

[15] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In *In Foundations of Genetic Algorithms*, page 1998. IEEE Press, June 2001.

[16] A. Gopalarathnam, B. A. Broughton, B. D. McGranahan, and et al. Design of low reynolds number airfoils with trips. In *19th AIAA Applied Aerodynamics Conference*, pages 2001–2463. AIAA E-library, June 2001.

[17] R. He and S. Sato. Design of a single-motor nano aerial vehicle with a gearless torque-canceling mechanism. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, pages 2008–1417. AIAA E-library, June 2008.

[18] R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15(1):407–412, 1978.

[19] J. H. Holland. Adaptation in natural and artificial system. Master's thesis, Stanford University, USA, 1992.

[20] B. R. Jones, W. A. Crossley, and A. S.Lyrintzis. Aerodynamic and aeroacoustic optimization of airfoils via a parallel genetic algorithm. In *7th AIAA/USF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 1998–4811. AIAA E-library, June 1998.

[21] I. Kroo and P. Kunz. The mesicopter: aminiature rotorcraft concept. Technical report, Stanford Univeristy, Palo Alto, 2000.

[22] B. M. Kulfan. A universal parametric geometry representation method-cst. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, pages 2007–62. AIAA E-library, June 2007.

[23] B. M. Kulfan and J. E. Bussoletti. Fundamental parametric geometry representations for aircraft component shapes. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 2006–6948. AIAA E-library, June 2006.

[24] P. J. Kunz. *Aerodynamics and design for ultra-low Reynolds number flight*. PhD thesis, Stanford University, USA, 2003.

[25] J. G. Leishman. *Principles of helicopter aerodynamics*. CambridgeUniv. Press, New York, 2005.

[26] W. Li and S. Padula. Performance trades study for robust airfoil shape optimization. In *21st Applied Aerodynamics Conference*, pages 2003–3790. AIAA E-library, June 2003.

[27] N. K. Madavan. On improving efficiency of differential evolution for aerodynamic shape optimization applications. In *10th AIAA/SSMO Multidisciplinary Analysis and Optimization Conference*, pages 2004–4622. AIAA E-library, June 2004.

[28] M. Nemec and M. J. Aftosmis. Aerodynamic shape optimization using a cartesian adjoint method and cad geometry. In *24th AIAA Applied Aerodynamics Conference*, pages 2001–3456. AIAA E-library, June 2006.

[29] A. Oyama and K. Fujii. Aerodynamic design exploration of flapping wing viewpoint of shape and kinematics. In *45th AIAA Aerospace Science meeting and Exhibit*, pages 2007–481. AIAA E-library, June 2007.

[30] S. Padula and W. Li. Options for robust airfoil optimization under uncertainty. In *9th AIAA Multidisciplinary Analysis and Optimization Symposium*, pages 2002–5602. AIAA E-library, June 2002.

[31] S. Peigin and B. Epstein. 3d optimization of aerodynamic wings based on accurate cfd computa-

http://www.imavs.org/

tions. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*, pages 2005–454. AIAA E-library, June 2005.

[32] G. M. Robinson and A. J. Keane. Concise orthogonal representation of supercritical airfoils. *Journal of Aircraft*, 38(3):580–583, 2001.

[33] J. A. Samareh. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA Journal*, 39(5):877–884, 2001.

[34] M. Secanell and A. Suleman. Sequential optimization algorithms for aerodynamic shape optimization. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 2004–4631. AIAA E-library, June 2004.

[35] M. S. Selig and M. D. Maughmer. Generalized multipoint inverse airfoil design. *AIAA Journal*, 30(11):2618–2625, 1992.

[36] M. S. Selig and M. D. Maughmer. A multi-point inverse airfoil design method based on conformal mapping. *AIAA Journal*, 30(5):1162–1170, 1992.

[37] H. Sobieczky. Parametric airfoils and wings. *Notes on Numerical Fluid Mechanic*, 68(1):71–88, 1998.

[38] B. I. Soemarwoto. Airfoil optimization using the navier-stokes equations by means of the variational method. In *16th Applied Aerodynamics Conference*, pages 1998–2401. AIAA E-library, June 1998.

[39] W. Song and A. J. Keane. A study of shape parameterisation airfoil optimization. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 2004–4482. AIAA E-library, June 2004.

[40] S. Sunada, T. Yasuda, K. Yasuda, and et al. Comparison of wing characteristics at an ultralow reynolds number. *Journal of Aircraft*, 39(2):331–338, 2002.

[41] C. H. Sung and J. H. Kwon. Design optimization using the navier-stokes and adjoint equations. In *39th AIAA Aerospace Sciences Meeting and Exhibit*, pages 2001–266. AIAA E-library, June 2001.

[42] S. Takahashi, W. Yamazaki, and K. Nakahashi. Aerodynamic design exploration of flapping wing viewpoint of shape and kinematics. In *45th AIAA Aerospace Science meeting and Exhibit*, pages 2007–481. AIAA E-library, June 2007.

[43] Noriaki Tsuzuki, Shunichi Sato, and Takashi Abe. Design guidelines of rotary wings in hover for insect-scale micro air vehicle applications. *Journal of Aircraft*, 44(1):252–263, 2002.

[44] H. Youngren, C. Kroninger, M. Chang, and et al. Low reynolds number testing of the ag38 airfoil for the samarai nano air vehicle. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, pages 2008–417. AIAA E-library, June 2008.

[45] Z .Q. Zhu, X. L. Wang, H. Y. Fu, and et al. Aerodynamic optimization design of airfoil and wing. In *10th AIAA/SSMO Multidisciplinary Analysis and Optimization Conference*, pages 2004–4367. AIAA E-library, June 2004.

http://www.imavs.org/

Propeller Location Optimisation for Annular Wing Design

Matthew Anderson[1,2], Kai Lehmkuehler[1], Derrick Ho[1], KC Wong[1] and Patrick Hendrick[2]

[1] The University of Sydney, Sydney, Australia
[2] Université Libre de Bruxelles, Brussels, Belgium
Matthew.Anderson@ulb.ac.be

Abstract

Most research today into the aerodynamics of ducts seeks to increase the thrust or efficiency of a propeller nested inside. This study takes a different approach to duct design and instead examines the use of a duct as an annular wing with the primary function of generating lift for high speed, cruising flight. The variables of investigation for this research are the location of the propeller inside the annular wing and how the thrust generated by the propeller effects the performance of the wing at various angles of attack. This investigation utilises wind tunnel testing to measure the forces generated by the wing and computational fluid dynamics to study the nature of the flow. The results show little difference aerodynamically between fore and aft mounting locations for the propeller in non-stalled flight, however, a forward mounted propeller is shown to be preferable when the flight envelope also includes the stalled regime. In hover, the aft propeller is shown to be favourable.

http://www.imavs.org/

Contents

1 Introduction **2**

2 Verification of Annular Wing Theory **3**

3 Experimental Set-up **4**

4 Computational Fluid Dynamics Set-up **5**

5 Results **6**
5.1 Experimental Set-up Verification 6
5.2 Effect of Propeller Location 7
5.3 Effects of Varying Thrust 8
5.4 The Effect of Windmilling 8
5.5 Static Thrust Tests 9
5.6 Flow Visualisation using Computational Fluid Dynamics 9

6 Conclusion **10**

7 Acknowledgements **11**

http://www.imavs.org/

1 Introduction

Annular wing aerodynamics (Figure 1 for notations), especially at low Reynolds numbers, is not a widely studied topic. Ribner [13] established the first firm basis for annular wing theoretical modelling, though the advantages of non-planar platforms were known much earlier from the works of Munk [7] and Prandtl [12]. Fletcher [3] conducted wind tunnel tests on five Clark Y profiled annular wings of differing aspect ratios from 0.33 to 3.0, providing one of the first experimental validations of the previous theoretical work of Ribner. Traub [17] furthered this by conducting experiments on two annular wings of aspect ratio 1.0 and 2.0, modifying the existing equations of Ribner for lift slope to take into account aspect ratio, as opposed to using two different equations for high and low aspect ratios. A further model was recently proposed by Maqsood and Go [6] that estimates the performance of annular wings using the leading-edge suction analogy.

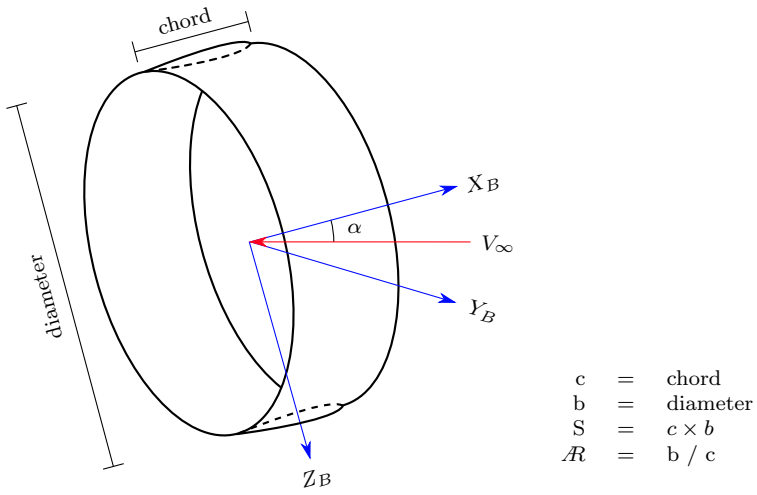


Figure 1: Annular Wing Notation.

The first to prove the advantages of ducted propellers experimentally was Stipa in 1931, and the concept grew in popularity with the development of VTOL and hover capable aircraft [14]. Today, their prevalence is only increasing, especially in the widely studied field of single ducted fan UAVs. Parlett [11] studied the effects of lip diameter, propeller rotation speed and Reynolds number and found that an increasing lip diameter increases the static thrust efficiency. Graf, Fleming and Ng [4] further studied the effects of lip diameter and found that though a larger lip radius enhanced thrust production in static conditions, a smaller lip radius performed more favourably in forward flight and in crosswind conditions. Yilmaz, Erdem and Kavsaoglu [18] investigated the effects of five different duct shapes and found the ducts could provide better propulsive efficiencies, observing up to 10 % efficiency increases, and the duct increased propulsive efficiency up until an advance ratio of 0.3. Their work also showed that a poorly designed duct, in this case only by varying the leading edge of the duct, could adversely affect the performance of the propeller. Sacks [14] notes that little work has been conducted on the position of the propeller within the duct, though he also states that the effect of propeller location could vary significantly in two different duct designs. A later theoretical study by Kriebel, Sacks and Nielsen [5] predicted that the position of the propeller has little effect on the performance of the duct in steady forward flight and hover.

2 Verification of Annular Wing Theory

Experimental investigations on annular wings have shown that theory, when correctly applied, can reasonably predict the lift slope of an annular wing. Experimental span efficiencies have been somewhat varied, although are consistently higher than those achieved by traditional planar wings.

Figure 2 shows a comparison of the theoretical models and wind tunnel experiments found in literature for the lift slope of annular wings [3, 6, 17]. Additionally, calculations using AVL [2] in the spirit of Traub [17] have also been included as a performance prediction method. While there is no 'one size fits all' analytical model for accurately predicting the lift slope, Traub's modified equation presents the best overall analytical prediction method, especially for aspect ratios of two and above. For lower aspect ratios, both the methods from Maqsood and Go and Lowry and Polhamus give very similar results, predicting the lift slope very well. AVL provides the best prediction method, though it is not a simple case of 'plug and play,' and requires some skill on the user end for correct utilisation.

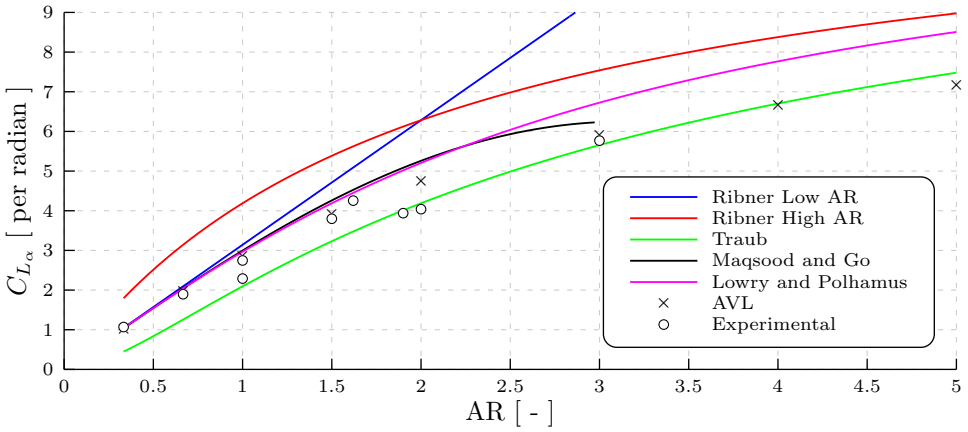


Figure 2: A Comparison of the Theoretical Models for Lift Slope and Wind Tunnel Experiments. Models from [2, 6, 13, 17], additional experimental data from [3, 6, 17].

Classically, the induced drag coefficient for annular wings is modelled as

$$K = \frac{1}{\pi \ AR \ e} \tag{1}$$

where the span efficiency, e, is two. Maqsood and Go proposed an alternate modelling of K such that

$$K = a\,(AR)^b \tag{2}$$

where a and b are curve fitted to existing data and equal 0.15 and -1.237 respectively. Equating Equations 1 and 2, the effective e as calculated by Equation 2 is found to be

$$e = \frac{1}{a \ \pi \ AR^{b+1}} \tag{3}$$

which causes some aspect ratio dependence on the span efficiency. Figure 3 shows a comparison of the theoretical models and wind tunnel experiments for span efficiency. Span efficiency is very sensitive to where the the induced drag is considered proportional to the lift squared, and the inherent inaccuracies of digitising data from graphs also affects the calculated value, thus it is very difficult to

http://www.imavs.org/

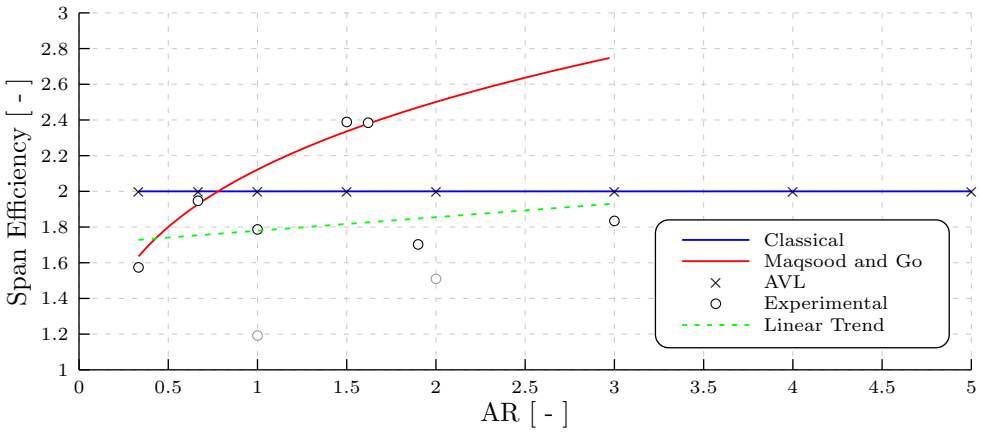
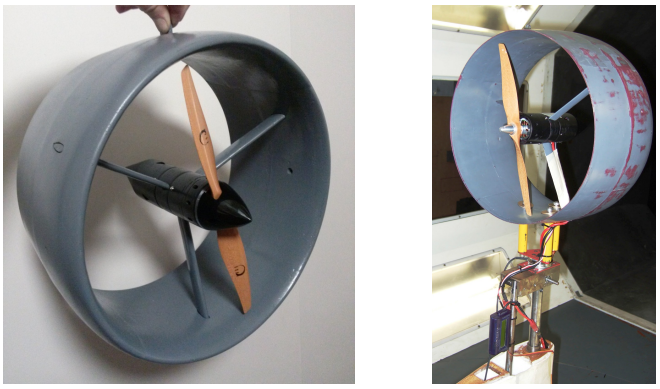


Figure 3: A Comparison of the Theoretical Models for Span Efficiency and Wind Tunnel Experiments. Models from [2, 6, 17], additional experimental data from [3, 6, 17].

get a definitive value from a data set. Nonetheless, the data tends to show values of around 1.8 being achieved experimentally, with little to no dependency on aspect ratio. The span efficiencies of Traub's data (the two greyed out data points) are lower than would be expected, though Traub theorises that this is most likely due to the mounting method of the the model in the wind tunnel.

3 Experimental Set-up

The experimental model (Figure 4) was fabricated using a Rapman Touch 3D printer using ABS plastic and finished to produce a smooth surface. The annular wing has a radius of 152 mm and a chord of 160 mm, giving an overall aspect ratio of 1.9. The aerofoil section is a NACA0012 and causes a slightly divergent inside section due to the aerofoil shape. The central hub section has a radius of 25.5 mm with a straight cylindrical section of 120 mm. The central hub is connected at the quarter chord point of the wing using three NACA0030 profile struts with a chord length of 20 mm.



(a) Tractor Configuration (b) Pusher Configuration

Figure 4: The Wind Tunnel Model with the Propeller Mounted in the Tractor and Pusher Positions

http://www.imavs.org/

The propeller is an 11x5 XOAR and can be mounted 20 mm from the leading edge of the wing for the tractor configuration, or 20 mm from the trailing edge of the wing for the pusher configuration. The clearance between the propeller and the wing is 2.9 % of the propeller radius for the tractor configuration and 6.8 % for the pusher configuration (due to the differing aerofoil thickness at each location). In order to calculate the forces of the wing and hub, thus excluding those of the propeller, the base drag as measured at zero angle of attack was assumed constant in each case, and any differences were attributed to the propeller and subtracted from the total forces. Two tests were run for each case, one at thrust equal to drag, 140 W for the pusher, 165 W for the tractor, and one at 300 W for both the pusher and tractor.

The wind tunnel experiments were conducted in the University of Sydney's '7x5' wind tunnel at 20 m.s^{-1}, corresponding to a Reynolds number of 1.94×10^5. The measured forces were corrected using the method from Shindo [15] and the downwash correction constants used were from Barlow, Rae and Pope [1]. Measurements were taken from -9 degrees in steps of 0.5 degrees until just after stall. The measurement accuracy of the load cell is ± 0.25 N, corresponding to a C_L and C_D error of 0.021, and there is a ± 0.5 degree measurement accuracy in the angle of attack. The values that were obtained in the '7x5' wind tunnel were also verified against earlier testing in the University of Sydney's '4x3' wind tunnel.

http://www.imavs.org/

4 Computational Fluid Dynamics Set-up

NUMECA FINE/Open with OpenLabs was used to model the wing and propeller set-up. The mesh consists of 2.6M cells and was generated using NUMECA Hexpress to create a fully hexahedral unstructured mesh. Only half the domain was modelled to save computational time and CPU booster and preconditioning were enabled to improve convergence times[1]. The Spalart-Allamaras (Extended Wall Function) turbulence model was used, and the boundary conditions for the turbulence were set from the recommendations of Spalart and Rumsey [16]. The perfect gas model for air was used to allow for compressibility effects and help the stability of the actuator disk. The boundaries were modelled as a velocity inlet / static pressure outlet system, enabling the effects of the propeller to be correctly modelled at the boundaries [9].

The propeller model used is a modified version of the actuator propeller model used by NUMECA's FINE/Marine package, implemented into FINE/Open using OpenLabs. The axial force profile is modelled such that

$$f_{b_{axial}} = \frac{105\ T\ \left(R_h/R_p - r/R_p\right)\left(R_h^2 - R_p^2\right)\sqrt{1 - \frac{R_h/R_p - r/R_p}{R_h/R_p - 1}}}{8\ R_p^2\ \left(3\ R_h/R_p + 4\right)\left(R_h/R_p - 1\right)^2} \tag{4}$$

where T is the desired thrust, R_p is the propeller radius, R_h is the hub radius, and r is the radial distance. The FINE/Marine propeller model also includes the ability to model the tangential force of the propeller such that

$$f_{b_{tangential}} = \frac{105\ Q\ \left(R_h/R_p - r/R_p\right)\left(R_h^2 - R_p^2\right)\sqrt{1 - \frac{R_h/R_p - r/R_p}{R_h/R_p - 1}}}{8\ r\ R_p^2\ \left(3\ R_h/R_p + 4\right)\left(R_h/R_p - 1\right)^2} \tag{5}$$

where Q is the torque and the other coefficients are as in Equation 4 [8]. These forces are then normalised by the volume of the actuator disk and are added into the appropriate momentum equations.

[1]Initial computational investigations on a full mesh showed that modelling the tangential force had little effect on the final total forces, thus were considered negligible and were not included in the final calculations.

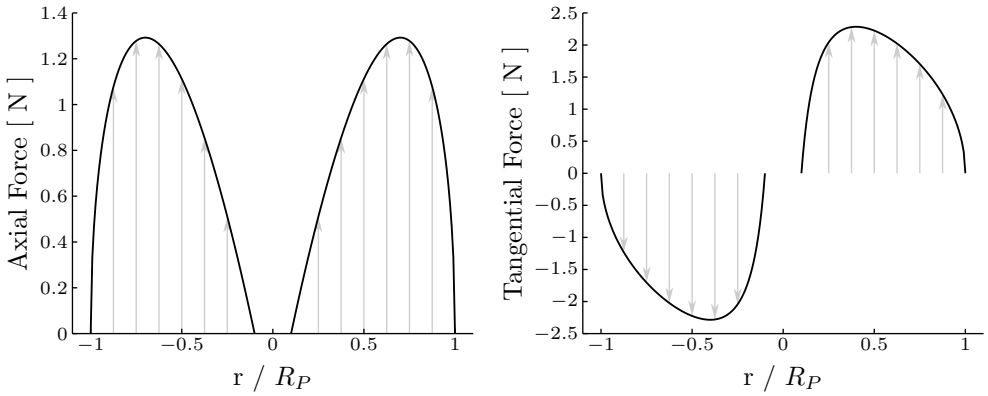


Figure 5: Axial and Tangential Forces for the Modified FINE/Marine Propeller Model

The loading profiles are given in Figure 5 for a propeller of $R_p = 1.0$ m, $R_h = 0.1$ m, $T = 1.0$ N and $Q = 1.0$ N.m. The OpenLabs implementation of the propeller model was verified against Froude's Propeller Theory in a standalone case to ensure the forces were correctly applied. For the computational investigation, three thrusts were investigated, 0.0 N, 2.5 N and 5.0 N, for both configurations.

5 Results

5.1 Experimental Set-up Verification

The results for the wing-only experimental tests (Figure 6) show good agreement with previous experimental data found in literature, verifying the experimental set-up ($\mathcal{A}R1.9$ data point in Figures 2 and 3). The lift slope is a lower than would have been expected, though this is most likely due to Reynolds number effects. The span efficiency for the wing-only is 1.7, while low, is within the expected range.

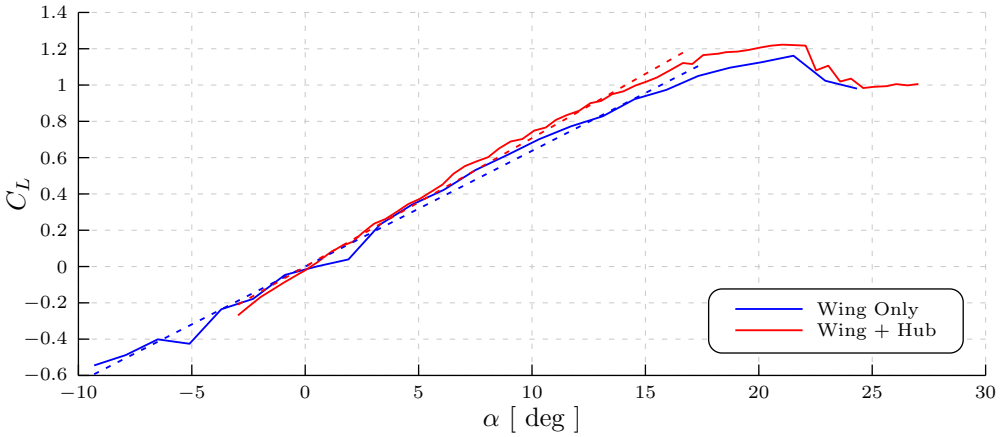


Figure 6: Coefficient of Lift for the Wing-Only and Wing and Hub Configurations. Trendlines dotted, Re $= 1.94 \times 10^5$.

http://www.imavs.org/

Adding the hub causes the experimental lift slope to increase slightly. The base drag significantly rises, though this is to be expected with the addition of a large body in the middle of the wing. The span efficiency is lowered dramatically by the addition of the hub to around 1.35. Finally, the neutral point shifts aft from $0.42c$ for the wing-only case to $0.68c$ for the wing and hub configuration.

5.2 Effect of Propeller Location

Figures 7 and 8 show the effect of the propeller at different power settings and locations on the lift and drag coefficients. Below stall, little effect is seen on the lift generated by the wing in any of the cases. There is slightly more effect on the drag characteristics, especially at higher power settings. The span efficiency is reduced for the tractor configuration, though the pusher configuration appears to have negligible effect on span efficiency. Interestingly, the power required to equalise thrust and drag at zero angle of attack for the pusher was lower than that for the tractor (140 W versus 165 W), resulting in 1.9 N and 2.2 N of generated thrust respectively. At 300 W, the pusher configuration produces 5.7 N of thrust, compared to the slightly lower 5.4 N for the tractor configuration.

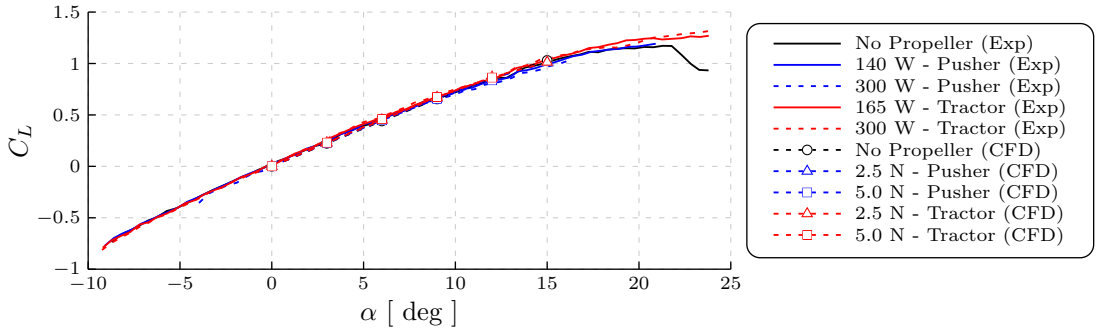


Figure 7: Experimental Lift Coefficient with Thrust Forces Removed and Computational Fluid Dynamics Lift Coefficient. Re $= 1.94 \times 10^5$.

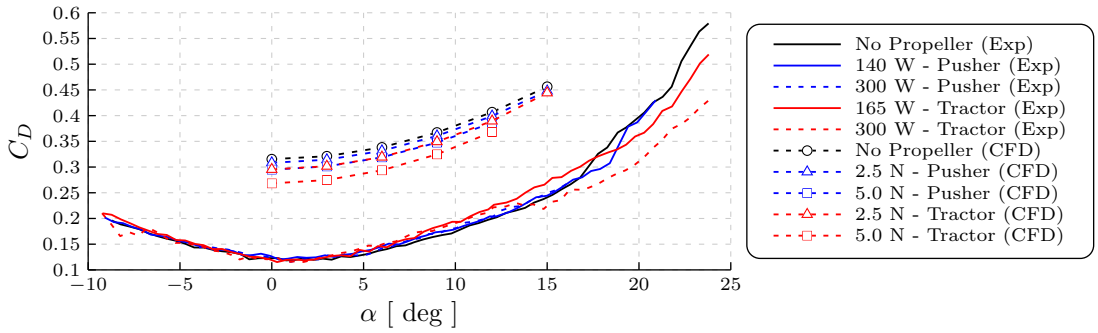
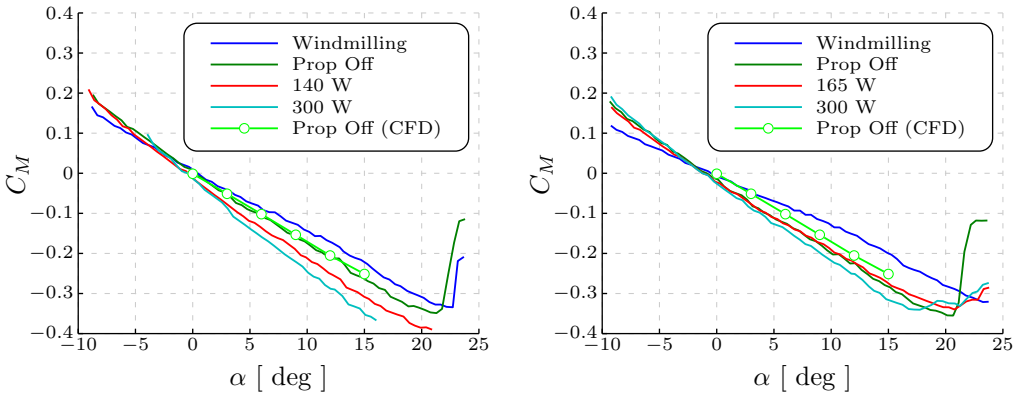


Figure 8: Experimental Drag Coefficient with Thrust Forces Removed and Computational Fluid Dynamics Drag Coefficient. Re $= 1.94 \times 10^5$.

Computational fluid dynamics (CFD) is shown to be able to predict the lift slope of the configurations very well below stall. While the base drag is not predicted correctly, the induced drag predicted

http://www.imavs.org/



(a) Pusher (b) Tractor

Figure 9: Pitching Moment Coefficient Variation with Propeller Increasing Power about 0.43c. $\mathrm{Re} = 1.94 \times 10^5$.

by CFD follows the experimental results. Futhermore, CFD suggests that the thrust from the propeller causes the base drag of the wing to decrease, especially in the tractor configuration. This can be attributed to the higher speed flow through the wing causing a low pressure region on the lip of the wing [10]. One important point of note is that the experimental results are conducted at a constant power, allowing for some change in generated thrust, while the CFD results are at a constant thrust. As such, the CFD provides an 'ideal' view of how the forces change with the propeller, whereas the experimental results give more of an idea of how this changes in the real world.

As the wing approaches and enters stall, larger variations between the configurations begin to show. The flow forced through the wing by the tractor configuration helps to delay stall by re-energising the boundary layer, increasing $C_{L_{max}}$ to 1.22 (from 1.05 for the no prop configuration) for the low power case. The pusher configuration does delay stall marginally as well, increasing $C_{L_{max}}$ to 1.12, though is far less effective than the tractor configuration. The 300 W case experienced heavy vibration at stall, and the $C_{L_{max}}$ gains at the lower power settings were heavily reduced or disappeared with increasing power. Stall has a greater effect on the pusher cases because the flow separation off the wing hits the propeller, causing a high degree of uneven loading, resulting in reduced propeller effectiveness and increasing vibration. The tractor propeller is in front of this separated flow, and thus doesn't suffer to the same degree as the pusher configuration in stalled flow.

5.3 Effects of Varying Thrust

Figure 9 shows the effect of increasing the thrust produced by the propeller on the pitching moment. As the propeller thrust is increased, the magnitude of the pitch moment slope increases, indicating the wing's neutral point shifts aft. Interestingly, the reverse happens for the windmilling case, shifting the neutral point forward. This is more pronounced for the pusher configuration, indicating that propeller suction has more of an effect on this phenomena than propeller slipstream.

5.4 The Effect of Windmilling

Windmilling causes the propeller to take energy out of the air, resulting in a force acting in the opposite direction to the powered cases. The 'thrust' was calculated as before, and was found to be

http://www.imavs.org/

-1.9 N for the pusher and -2.5 N for the tractor. Subtracting this force from the recorded forces results in little change in the below stall C_L and C_Ds from the unpowered wing, much like for the powered cases. Windmilling also causes a decrease in the $C_{L_{max}}$, reducing it to 1.00 (from 1.05) in both cases. The violent stall behaviour experienced in the powered tests was also present in the windmilling tests, and was particularly strong for the pusher configuration.

5.5 Static Thrust Tests

Figure 10 shows the net static thrust (as for a hover flight condition) of the pusher and tractor configurations at several power settings. Though the tip clearance is higher for the pusher configuration, it appears to generate around 7 % more thrust at 300 W. This is possibly due to the wing acting as a guide vane, providing cleaner airflow into the propeller. Unlike the tractor configuration, the pusher propeller also sits behind the hub mounting arms resulting in no slipstream blockage.

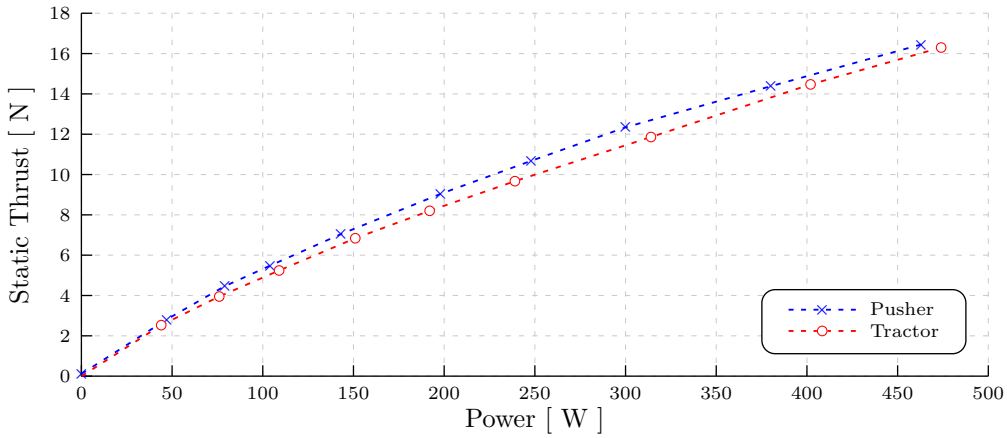


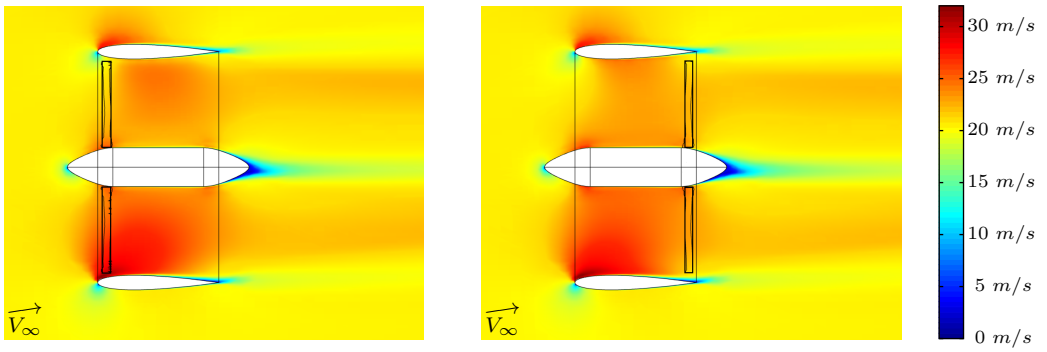
Figure 10: Net Static Thrust Comparison between Pusher and Tractor Configurations. Re = 1.94×10^5.

5.6 Flow Visualisation using Computational Fluid Dynamics

Computational fluid dynamics allows a constant amount of thrust to be applied, regardless of the incoming flow. This allows for the study of the forces over the wing regardless of propeller performance. This is in contrast to the experimental investigation which was carried out with constant propeller power, which allows for some degree of varying thrust.

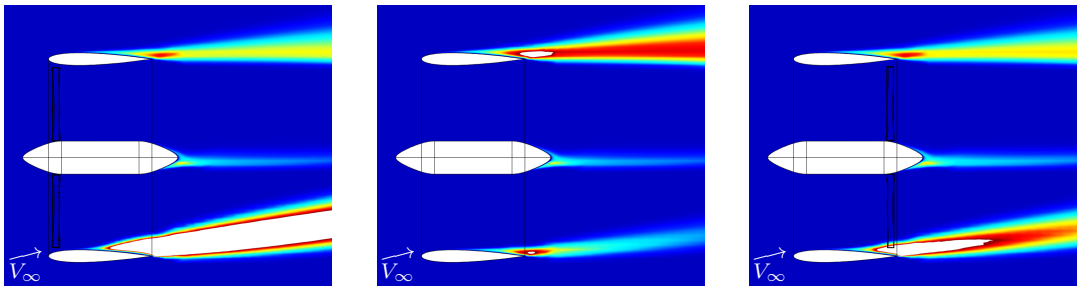
Figure 11 shows the magnitude of the velocity at a cut through the ZX plane of the wing at 9 degrees angle of attack and in non-stalled flow. Though the tractor propeller sits at the front of the wing, it experiences uneven loading between the top and bottom of the sweep due to the influence of the wing on the airflow (Figure 11a). This causes a decrease in effectiveness of the propeller to generate thrust, and can also cause unwanted vibrations in the aircraft. For the pusher configuration, by the time the air reaches the propeller at the back of the wing (Figure 11b), it has become much more uniform, allowing the propeller to generate thrust far more efficiently. In addition to this, the boundary layer is well developed when it reaches the pusher propeller, which reduces the effective gap between the wing and the propeller, reducing tip vortices leading to lower power consumption.

http://www.imavs.org/



(a) Tractor (b) Pusher

Figure 11: Magnitude of the Velocity for the Tractor and Pusher Configurations in Non-Stalled Flow. Angle of Attack = 9 deg, T = 2.5 N, Re = 1.94×10^5.



(a) Tractor (b) Wing (c) Pusher

Figure 12: Turbulent Viscosity Showing Differing Stall Characteristics between Each Configuration. Angle of Attack = 15 deg, T = 2.5 N, Re = 1.94×10^5, scale from blue ($\mu_t/\mu = 0$) to red ($\mu_t/\mu = 120$).

Figure 12 shows the effect of the propeller on the wing as stall is approached. Traub [17] noted that the top section of an annular wing stalls first, a phenomenon evident in Figure 12b. Figures 12a and 12c show the lower section stalling first as a result of the extra airflow forced through the wing. Furthermore, Figure 12c shows the pusher propeller is already starting to ingest highly turbulent flow, while the tractor propeller (Figure 12a) is still able to operate in relatively clean air.

6 Conclusion

This study shows that the position of the propeller inside an annular wing makes little difference to the steady state, non-stalled forward flight performance, though there is evidence to suggest that the pusher configuration requires less power to remain in steady flight than the tractor configuration. The presence of the propeller at moderate power is shown to delay stall, increasing $C_{L_{max}}$, though too much thrust can cause degradation in the performance of the wing. With increasing thrust, the aerodynamic centre is shown to shift aft.

As stall is approached, the tractor configuration is shown to be preferable, not suffering from ingesting the detached flow of the wing like the pusher configuration. Static thrust tests show the

http://www.imavs.org/

pusher configuration to be preferable, producing more thrust for a given power despite a higher propeller clearance than the tractor configuration.

These tests were only conducted for one type of propeller and more investigation is required to ascertain if the propeller choice had an effect on the results. Additional variables not investigated here, such as the placements of guide vanes and control surfaces, should also be considered and may have a more profound effect than propeller placement in steady state, non-stalled forward flight.

7 Acknowledgements

The authours would like to thank Blake Formatti for the initial design and fabrication of the wind tunnel test model, Frank Buysschaert for his technical input and NUMECA for their support.

References

[1] J. Barlow, W. Rae, and A. Pope. *Low Speed Wind Tunnel Testing*. Wiley, 3rd edition, 1999.

[2] M. Drela. AVL. `http://web.mit.edu/drela/Public/web/avl/`.

[3] H. S. Fletcher. NACA Technical Note 4117: Experimental Investigation of Lift, Drag, and Pitching Moment of Five Annular Airfoils. Technical report, National Advisory Committee for Aeronautics, October 1957.

[4] W. Graf, J. Fleming, and W. Ng. Improving Ducted Fan UAV Aerodynamics in Forward Flight. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, 2008.

[5] A. R. Kriebel, A. Sacks, and J.N. Nielsen. Theoretical Investigation of Dynamic Stability Derivatives of Ducted Propellers. Technical report, Bureau of Naval Weapons, US Navy, 1965.

[6] A. Maqsood and T. Go. Aerodynamic Estimation of Annular Wings Based on Leading-Edge Suction Analogy. *AIAA Journal*, 51(2):529–534, February 2013.

[7] M. Munk. NACA Report No. 151: General Biplane Theory. Technical report, National Advisory Committee for Aeronautics, 1923.

[8] NUMECA International. *User Manual: FINE/Marine v2*, March 2012.

[9] D. O'Brien. *Analysis of Computational Modeling Techniques for Complete Rotorcraft Configurations*. PhD thesis, Georgia Insititute of Technology, 2006.

[10] O. Ohanian. *Ducted Fan Aerodynamics and Modeling, with Applications of Steady and Synthetic Jet Flow Control*. PhD thesis, Virginia Polytechnic Institute and State University, 2011.

[11] L.P. Parlett. Aerodynamic Characteristics of a Small-Scale Shrouded Propeller at Angles of Attack from 0° to 90°. Technical Note TN3547, NACA, 1955.

[12] L. Prandtl. Induced Drag of Multiplanes. Technical Note 182, NACA, 1924.

[13] H. S. Ribner. The Ring Airfoil in Nonaxial Flow. *Journal of Aeronautical Sciences*, 14(9):529–530, 1947.

[14] A. Sacks and J. Burnell. Ducted Propellers - A Critical Review of the State of the Art. Technical report, Advanced Research Division of Hiller Aircraft Corporation, 1959.

[15] S. Shindo. Simplified Tunnel Correction Method. *Journal of Aircraft*, 32(1), 1995.

[16] P. Spalart and C. Rumsey. Effective Inflow Conditions for Turbulence Models in Aerodynamic Calculations. *AIAA Journal*, 45(10):2544–2553, 2007.

[17] L. W. Traub. Experimental Investigation of Annular Wing Aerodynamics. *Journal of Aircraft*, 46(3):988–996, May-June 2009.

[18] S. Yilmaz, D. Erdem, and M. Kavsaoglu. Effects of Duct Shape on a Ducted Propeller Performance. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013.

http://www.imavs.org/

Technical Session 2

Mission for MAVs

Chairman : Peter Vörsmann

- Vincent Van Geirt and Renaud Kiefer. ***Endurance UAV glider for topography application***
- Gautier Hattenberger, Grégoire Cayez and Greg Roberts. ***Flight tests for meteorological studies with MAV***

http://www.imavs.org/

Long-Endurance UAV Glider for Topography Application

Vincent van Geirt[1]*and Renaud Kiefer[2]†

INSA Strasbourg, Strasbourg, France
vincent.vangeirt@insa-strasbourg.fr
renaud.kiefer@insa-strasbourg.fr

Abstract

We are actually developping a long-endurance glider. The aim of this application is to create an UAV for a topography application. We want to recreate by computer way a site (200*200m) in 3D. This could only be done with special photography of the zone. These photos will be taken by our drone in an altitude of 300m. We had to develop an extra-light glider with high aerodynamics performance to ensure an automatic flight of 1h. In the following paper, we will explain the design, the CAD and composite conception and the choice we have made to create it and how it is electronically controlled.

http://www.imavs.org/

1 Introduction

This project has been developped in the INSA Strasbourg's school, a french engineering school based in Strasbourg. It last two years and the aim is to design and create a long-endurance UAV for topography application. The team is composed by 6 students. The work presented in the following pages has begin 18 months ago and is part of the mechatronic cursus (4 hours per week).

This paper will be devided such as presented below :

1. First of all, the constraint of our application will be explained.
2. The specifications, which results from the constraint and some choices.
3. The mechanical part of the UAV especially the fuselage and wings.
4. The electronic design, the choice of the component and the control will be explained.
5. The current state of the project and the results of first flight will be commented, and we will give the perspective of the project in the following months.

2 Constraint

This project is developped for a topography application because it is a specialization at INSA Strasbourg. It is born of a need : having a long flight to recreate 3D landscapes where the topographs could eventually not placed their devices. The plane must be able to hover a field, to take pictures of it but also to travel from a site to another to ensure a good landing for example.

*team leader of the project
†supervisor of the project

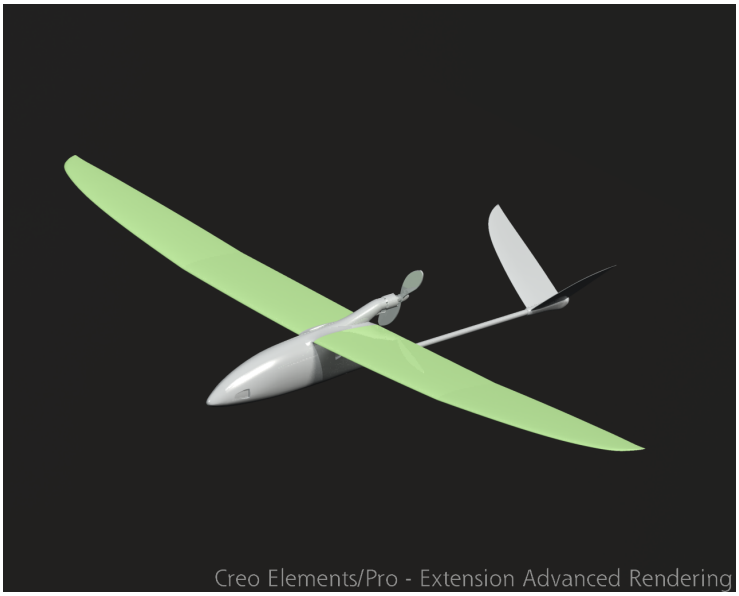


Figure 1: glider's CAD

The goal is to cover a 200m*200m zone. The spot has to be captured with pictures which recovered themselves for 60% in width and 30% in height. The minimal resolution is about 1 pixel for a 10*10cm in the field. The camera has to be always parallel to the ground ($\pm$ 5).

3 Specifications

With the constraint explained in the last section 2, the specifications have been built.

To ensure to have enough energy to flight in and out the zone, we have fixed a minimum flight of 1 hour. The wingspan has been fixed to 3m. The weight of the system has to be lighter as possible. The wing loading has been chosen of 4kg. This data is the consequence of the weight of the battery and the motor we want to board in, and of an estimation of the structure weight, plus a security margin.

Besides that, we need to have a very high resolution for the vision, and a safety mode in case of emergency to avoid the crash of our UAV.

3.1 Dimensionning the motor and the battery

To calculate it, we have based our calculation on the Newton's second law of motion. The result of this equation is :

$$P = \frac{\rho_{wind} * S * C_x * V^3}{2} + m * g * \sin(\alpha) * V \quad (1)$$

http://www.imavs.org/

with :

- P the power,
- ρ_{wind} the wind density
- S the wing surface (0.82kg/dm^2)
- C_x the drag coefficient
- V the speed (km/h)
- m the weight of the system (4kg)
- g the gravity
- α the angle between the plane and the ground in degree (null when the flight is horizontal)

By placing the speed and the angle α as parameters we have found the optimum to have more power by using the less energy.

The battery has been calculated with this optimum to ensure to have enough energy for the long-endurance flight. The weight of the battery highly impact the weight of the whole system, we have to verify if we do not exceed the total weight of the system with the battery (according to the specifications section 3 4kg).

The optimization reveal that it is better to take off with a "high" angle but during a short period rather than with a "small" angle during a long time. Take off is made with $\alpha = 35$. This optimum has been found with the following value as parameters :

S (kg/dm^2)	C_x	V (km/h)	take-off time (s)	α (^{o})	total power (W)
0.82	0.02087	54	17.43	35	626.5

The values has been established with a motor efficiency of 0.88 and a propeller efficiency of 0.6. The notation corresponds to those introduced for the equation 1

The chosen elements are :

battery : 2 will be placed in parallel to provide enough energy : 4000mAh, 40C each.

motor : brushless motor with a nominal power of 800W.

http://www.imavs.org/

4 The Mechanical Parts of the UAV

4.1 Choice of the camera

As exposed in section 2, there is a need to have a high resolution and a very light weight for the camera. The criteria of comparaison were mainly the resolution, the weight and the price. A ranking of the market solution has been made. To be short, in this paper, we will not give all the possibilities we have studied but just the one we have finally chosen : the GoPro.

Its advantages are :

- the weight : 100g
- the resolution is good : 1080p
- It can take until 30 images/s (film mode)

Nevertheless, it also has its disadvantages, the main one is its fish-eye effect. This effect is really annoying in the corner of the picture. The center is not really distorded. After a quantification of the distorsion with a line of sight, a crop will be necessary. The original images taken are 1920*1080 and after the operation, the resolution will be : 1600*900.

This gives the result of 1px in the image correspond to 6*6cm in the field at the desired altitude : 300m.



Figure 2: crop made around the non-distorded zone, image taken during a flight test

4.2 Fuselage's study

We have thought the fuselage to have the best access to the inside component, and naturally to have the smaller drag force as possible.

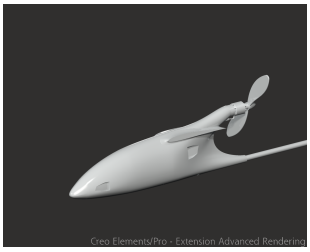
The original idea was to observe the nature. Wich form is the most aerodynamics ? It reveals that the drop was the best one. So we begun to draw a fuselage close to this shape. After some fluid analysis, comparison between different shapes the final shape has been made.

http://www.imavs.org/

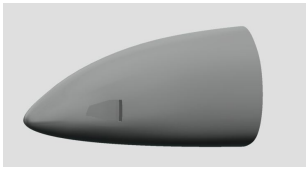
It caracteristics are the following :

- drag force : 2.5N
- estimated weight : 300g with composite material (Carbon,Kevlar)
- total length (with the tailplane) : 1370mm
- height : 210mm
- width : 100mm

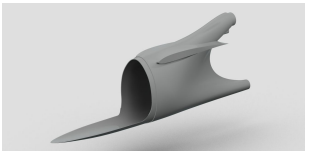
It is made by two parts (as shown below) : the body which will be holding the inside component and a head to protect them against the outside conditions.



(a) View of the closed fuselage



(b) View of the head



(c) View of the body

Figure 3: CAD of the fuselage and its parts

Because of the heat loss of the battery, we have to designed airing. We have chosen the normalised NACA inlet and outlet. To ensure a correct flow, the outlet NACA is 2.5 bigger than the inlet one. Its size has been dimensionned with an estimation of the loss due to the Joule effect.

4.3 Wings' study

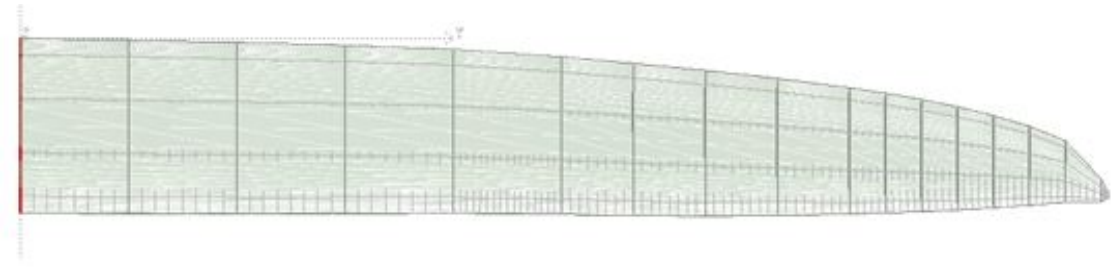
The wings are inspired by F3F and F3B competition models, because theirs performances are enhanced to have the highest lift-to-drag ratio. This ratio is for the application also very important because of the will to have a long flight.

The shape is pseudo-elliptic, there is a diaedral of 1 at the butt and 4 at the middle. In order to have the best output, the wings has been design with anhad an elliptic lift distribution (evolution of the airfoils shape along the wing). This has been tested with XFLR5 under two incidences : 0.5 and 1. We have only tested these values because we thought that they were the optimal one. The graphics 4b shows the results.

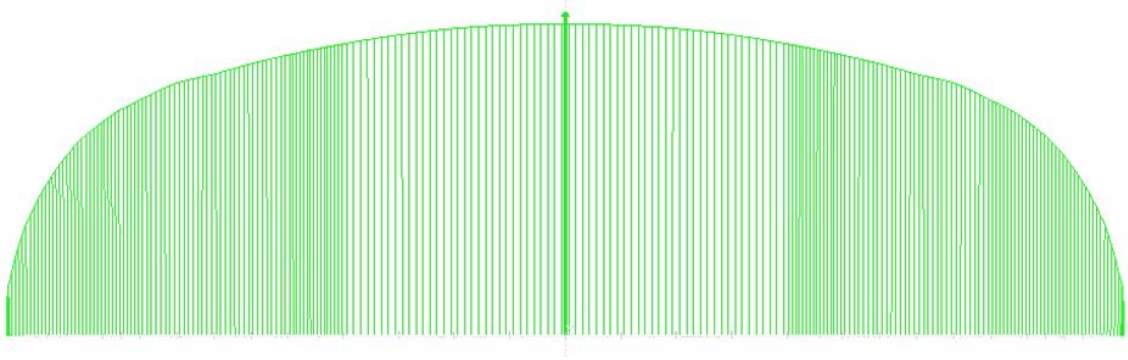
We have used the XFLR5 software to determine the characteristics of the wings :

- lift-to-drag ratio maximal with an incidence of 0.8
- The power factor is maximal with an incidence of 1.2
- The lift coefficient of 0.1 at an incidence of 1

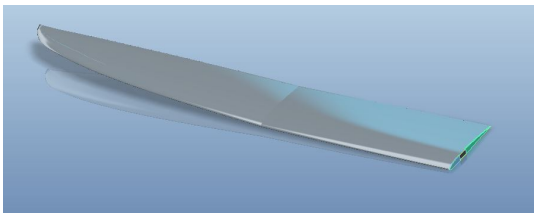
http://www.imavs.org/



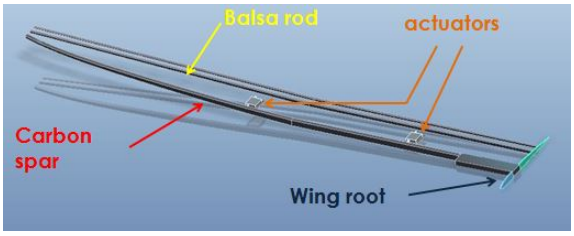
(a) XFLR5's modelisation



(b) lift distribution of the wings (Cz vs wingspan)



(c) CREO's modelisation of the wing



(d) CREO's modelisation of the inside of the wing

Figure 4: CAD of the wings

These simulations have helped us to the elaboration of an aerodynamic model of the flight. This is useful to enhance the control. This model is actually under development.

The wings have been modelized in a CAD software to integrate the actuators inside.

The wings are made of a sandwich composite material : Carbon - Airex - Glass fibers. The carbon and the glass fibers are used in flexion and the airex is placed to rigidify the structure in traction and compression. We have also placed a carbon spar and a balsa rod to rigidify the whole wing.

http://www.imavs.org/

4.4 Realisation of the fuselage and the wings

The choice of a composite structure has been made because it combines a light weight and good mechanical properties. But it also means that the parts have to be moulded. The wings have been made by contact molding. To do so, we made a wood master, this master have been moulded to create the mould. Then the material have been put layer by layer to make the wing.

To make the fuselage, the same steps has been required. Actually, the master has been made and the mould will be create very soon.

5 Electronics Design

5.1 Electronic components

To provide an automatic fligt, we are using the following component :

- The motherboard is a DSPIC33F
- Three axis accelerometer to determine the angular velocity of the plane
- One gyroscope for each axis
- 4 servomotors as actuators to move the aileron
- 2 servomotors in the fuselage to control the stabilization's aileron
- 1 servomotor to trigger the camera
- 1 brushless motor
- 1 GPS to acquire the position of the plane and a first value of the altitude (incertitude : 2.5m[1])
- 1 barometer to acquire the altitude (incertitude 1m[1])
- 1 telemeter to have a precise altitude measure for take off and landing (incertitude : 0.01m[1])
- 1 GoPro to take the pictures
- 1 radiocommand and its controller
- 1 xBee module to ensure the communication to the ground during flight.

During the testing phases and to allow developping the electronic while the fuselage is not moulded, we use a polystyrene glider. Every component are integrated in it and we can test the functionning of our electronic. To allow a quicker development we have also used the UDB4 of SPARKFUN[4] : it is a development board which contains the gyroscope and accelerometer. Part of the code we have implemented can be found in gentlenav [2].

[1] this incertitude has been experimentally verified

http://www.imavs.org/

5.2 Control

The glider has four flight mode :

manual : There is no control of the output. The board take the PWM from the RC receiver and transmit it to the actuators without treating the information. This mode needs a pilot. It has been developped in emergency case, to avoid a crash of the UAV. It has the priority to any other mode.

semi-automatic : In this mode, there is still the need of a pilot, but the plane is regulated in angular speed. It means that the plane is forced to stay in an angle cone for roll and pitch. Yaw angle is not critic to avoid a crash so this angle is not controlled. This mode was a first step in the development of the control. It can be use by unexperimented pilots to avoid a crash. The angles has been chosen to be harmless for the plane ($\pm$ 30). The most interesting part of this control mode is that it correct automatically its trajectory when an external perturbation come.

automatic : This mode is the most important for our final aim. It allows an automatic flight without pilot. The goal is that the glider take off on his own, go on the wanted site, take the pictures of the zone and come back with an automatic landing. To do that, the waypoints have to be program before the flight. The yaw, pitch and roll are electronically controlled with the informations of the sensors. It will be developped in the next section.

return to launch : If the glider lost the RC contact, it has stored the initial position in its memory and goes back on its own to the initial point, where it could land by circling down until the ground.

To switch from one mode to another, a switch on the RC controller has been program. With three positions, we can easily change the flight mode if there is a problem.

5.3 An automatic flight

The control is made by the DSPIC33F. It receives the information for the gyroscope and accelerometer, these informations are averaged, if some are too far from this average they are not taking into account. The state of the plane is determined with the DCM algorithm of gentlenav [2]. This algorithm update the 3*3 Matrix with the informations of the sensors.

Depending on the goal the pane has to reach (the next waypoint), it corrects if necessary its trajectory. The acquisition of this parameter is made at 8Hz. It means that we have 8 sets of data per seconds. That allows the control to be reactive to an external perturbation.

The control is made by a PID controler on each axis. The coefficients of these controlers are, for the moment, empirically found. We are currently working on a Matlab model of the plane to optimize the control coefficient. This work is not yet finish so I will not present it in this paper.

The waypoints are coded in the program and we can change it by reprogramming the part of the program which contains it. We can also add some waypoints during the UAV flight. To do so, we have a link to the ground with an xBee module, this is the object of the following part.

To code the waypoints we have two possibilities, to indicate it with the GPS coordinate (longitude, latitude) in degrees and altitude in meters or we can also program it relatively to the points where the UAV has been started. Indeed at the start of the glider, an initialization of all the sensors is made.

http://www.imavs.org/

The initial point is given by the GPS and is stored in the memory. So we could indicate the waypoints relatively to this point by indicating the distance toward East, North and the altitude in meters. This is usefull to make a route without searching the exact GPS coordinate.

5.4 A link to the ground

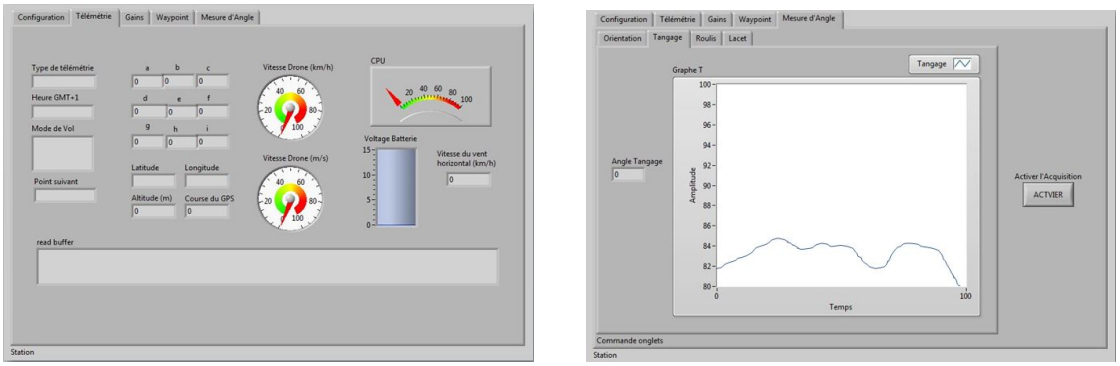
The xBee Pro module is a device which allows serial data transfer with a range of 1500m [1]. Datas can go both ways, from the computer to the plane and vice versa. We receive the data from the location and orientation of the plane at 8Hz. This data are send to an interface in Labview which returns the position, speed and the battery level of the plane. It also returns, with these datas, graphics of the last orientation of the UAV. We can follow it by seeing these graphics.

We also used an interface developped in open source found on the internet called HK Ground Control Station [3]. This interface needs to have an internet connection to work but it allows a map visualisation of the drone with the same data as ours. The altitude is symbolised with a parallelogram which has a bigger side according to the altitude. It is very convenient but the one developped with labview has the advantage to work without an internet connection and to provide graphics of the angle evolution which is usefull to optimize certain parameter empirically. Some snapshot of these interfaces are displayed below, figures 5a and 5b .

On the HK figure 6, the red circle shows the place of the waypoints in a test flight done in april. This shows that the UAV goes throw the waypoint. Between, if there is wind or external perturbation, it will only correct its trajectory to be on the next waypoint. To have a precise trajectory, it needs to have a lots of waypoint not far away from each other (30 to 50m). Here in the picture, the waypoints are separated by 200m from one another.

The xBee module allows both-ways data transfer so we have an interface to send some datas to the UAV. This is used to change the coefficient of our PIDs controllers. Indeed, actually they are empirically found, we can change it on flight. It is more convenient than reprogramming all the device. We can also send waypoints to the UAV. This waypoint will be add in place of the last and will be taken at the end of the other waypoints. These information are send at 1Hz, so every second we can send a data set of information to either modify the reactivity of the UAV or add waypoint to the route.

http://www.imavs.org/



(a) data page of our interface (b) roll's graphics

Figure 5: Labview Interface



Figure 6: HK Ground Control Station

6 Current State of the Project

There is still six months of work on the project, so everything is not yet finished. The points beneath resume the achivement and the following to come.

mechanical part : The wings are now produced, we are currently making the mould for the fuselage. Once it will be finished, we have to mould the fuselage with the composite fibers. Then all

http://www.imavs.org/

components could be integrate in it and a test flight with the final structure could be done (all the actual test flight has been made with a polystyren structure).

electronical part : All the three flight modes are functionnal, we are currently working on the integration of the barometer, the altitude is given only by the GPS for the moment, the incertitude is too big. The telemeter is functionnal but the lack of barometer made the landing impossible. Once it will be functionnal, we will integrate the landing in the automatic mode. We are also working on a model of the flight to optimize it.

7 Conclusion

In this paper, we have tried to synthetise the work done by a team of students in the INSA Strasbourg's school. The main point of our application are the long-endurance and the acquisition of images automatically for the topographs. For the first point, it means a big battery to provide enouh energy for the whole flight and a design which allows to have very low friction and asperity to be the most aerodynamic as possible. The shape and the roughness of the UAV are very important. For the second one, we have worked on the control of the drone to allow an automatic flight, and also on the interface to permit a visualisation of the flight. This visualisation is usefull to detect the problems and the possible improvement. The specificity of our project is that it is entirelly develop by engineering students. From the design of the prototyp to the realisation of it.

Thanks :

The team wants to specially thanks the IMAV 2013 edition to allow us to present our work.

The INSA Strasbourg *'s school, for its support in this project and more particulary M. Kiefer and Vedrines, supervisors of the project.*

Friends and collegues who have helpfully contributed to the project, especially :

Florian Saint Leger : *For the conception and realisation of the fuselage.*

Miao Lin : *For his contribution to this project despite the difficulty of the french language !*

Aubrie Callies : *For her work of modelisation of the wings and also her preparation for the moulding.*

Clementine Astier and Maxime Schunder : *For the realisation of our wings and their expertise in composite material.*

Pia Jacobi : *For the work in the electronic control, thanks for never give up !*

Florian Breut : *For his huge work of calculation and his interface*

Ulrich Nsenguet : *For his particular work on the ground station.*

All those who have contributed to this project.

http://www.imavs.org/

References

[1] Digi. datasheet of the xbee-pro module. Available at https://www.digi.com/pdf/ds_xbeemultipointmodules.pdf.

[2] Bill Premerlani. Firmware for bill premerlani's imu based uav dev board autopilots. Available at https://code.google.com/p/gentlenav/.

[3] pualbmather@gmail.com. Happykillmore's ground control station. Available at https://code.google.com/p/happykillmore-gcs/.

[4] sparkfun.com. Udb4 - pic uav dev board. Available at https://www.sparkfun.com/products/retired/10582.

http://www.imavs.org/

Flight tests for meteorological studies with MAV

Gautier Hattenberger[1], Grégoire Cayez[2] and Greg Roberts[3]

[1] ENAC, Toulouse, France
gautier.hattenberger@enac.fr
[2] ENM, Toulouse, France
[3] CNRM-GAME, Toulouse, France

Abstract

The use of UASs (Unmanned Aerial Systems) in meteorological studies has been a growing interest in the recent years but several technical and legal issues have to be overcome in order to fly safely and efficiently within the civil airspace. This paper addresses the technical aspects of the VOLTIGE (Vecteur d'Observation de la Troposphère pour l'Investigation et la Gestion de l'Environnement) project where a fleet of UAVs will fly simultaneously in and out of fog. Several improvements have been made on the communication between the payload and the autopilot in order to have an efficient sensor-based navigation. The challenges regarding the integration into civilian airspace are also presented.

http://www.imavs.org/

1 Introduction

The use of UAVs in meteorological studies has been a growing interest in the recent years [5, 1, 6]. Several types and sizes of platforms have already been used around the world with different technical and legal issues. Small UAVs have shown especially good results in observing the planetary boundary layer [4, 3]. When studying fogs, most of the measurements are currently constrained to ground-based observations or vertical soundings. The use of manned aircraft is too dangerous in this particular situation. Small and micro UAVs are probably the best tool to observe fog events since they can safely fly close to the ground, yet adapt their trajectory to the current meteorological conditions. Such a sample strategy has not been permitted before.

This paper will first introduce the main objectives of the VOLTIGE project. The initial and improved architecture of our autopilot and payload control are then described (including a justification of the tight integration between payload and navigation systems). Finally, the flight campaigns that have already been carried are presented with some preliminary scientific results. The regulatory aspects concerning those flights are addressed as well.

2 Objectives

One of the VOLTIGE project goals is to perform a coordinated flights of several UAVs equipped with meteorological sensors for the study of fogs. These flights have to be performed within the civil airspace while complying with the French airspace regulations. Another key aspect of this project is the improved integration of the communication between the payload and the autopilot in order to have an efficient sensor-based navigation.

The meteorological sensors include pressure, temperature, humidity, solar radiation, cloud and fog detection, and turbulence. Three types of flights' profiles will be performed simultaneously from different aircraft in order to gather the scientific data:

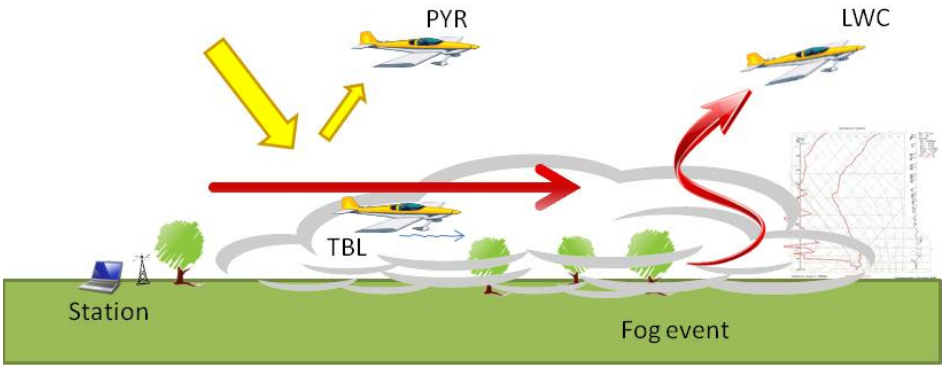


Figure 1: Principals of the VOLTIGE project fog study

- turbulence measurements from an horizontal flight inside the fog (TBL on Figure 1),
- vertical profiles from ground level to above the fog (LWC),
- radiation measurements above the fog (PYR), which implies detection of the fog boundaries.

Since the visibility will be reduced, it is important that the aircraft can perform automatic collision avoidance. This safety system is based on the same principals than the TCAS [2] used on commercial aircraft.

3 System architecture and integration

3.1 Initial setup

The autopilot system used in the project is the Open-Source UAV system *Paparazzi*[1]. The initial setup for this project includes two independent inboard electronics. The first part of the system consists of the flight control system, which is composed of the main autopilot board, a GPS, a radio modem, a RC receiver for the safety pilot, a motor and the actuators. The second part is the scientific payload (Fig. 2) including the sensors and specific electronic boards (usually two boards in each flights and a data logging system with its SD card. Each part has its own power supply.

While offering a fast and easy integration for the preliminary flight tests and sensor calibration, this solution has a lot of limitations and drawbacks, such as difficulties to fit all the components in a reduced space, redundant capabilities between the mission payload and UAS leading to an unnecessary increased weight, EMI issues,...

3.2 New autopilot development

In order to overcome the difficulties arising from separate electronics while offering required capabilities for atmospheric studies, a new autopilot hardware design has been initiated with the following constraints: payload management and sensor reading on the main autopilot board, data logging, and small size.

[1] http://paparazzi.enac.fr

http://www.imavs.org/

http://www.imavs.org/

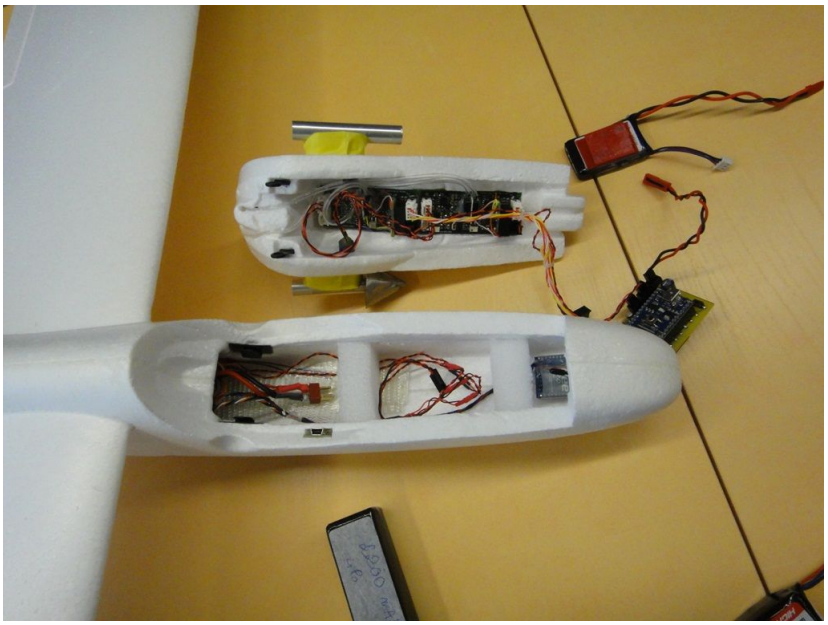


Figure 2: Payload electronics for the initial setup

The result is an autopilot, named *Apogee* (Fig. 3, right), with the same size as the previous generation, with a micro-SD card slot and more powerful micro-controller unit capable of handling the autonomous navigation of the MAV, a wider range of payload with more inputs/outputs, and high speed data logging (see characteristics in table 3.2).

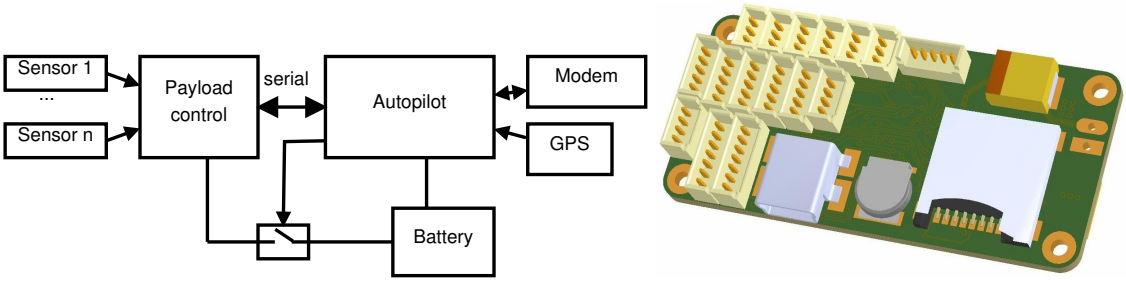


Figure 3: Integrated architecture and new autopilot design

This integrated architecture (Fig. 3, left) will allow to easily exploit the scientific data from the payload sensors in order to optimize the flight plan. Since the goal is to scan a wide and unknown volume, the trajectory of the aircraft has to be dynamically adapted to the meteorological conditions to gather a maximum of useful data. In order to optimize the battery consumption, the autopilot can control the power supply of all the payload and activate them only during the measurement phase of the flight.

With this architecture, the payload controller is able to either send commands to the autopilot, such as navigation commands or meteorological information (e.g. fog's boundary detection), or send

Processor	STM32F405
IMU	MPU-6050
Barometer	MPL3115A2
Logging	MicroSD slot over SDIO interface
Connectivity	6 PWM, 3 UART, 2 I2C, 1 SPI, 1 SWD, 1 USB, 4 AUX (PWM, ADC, GPIO)
Remote control	1 PPM input, 1 serial input
Power	1.5A switching power supply, 1 5V/500mA power switch
Size x weight	53x25 mm, 10.4 grams, 4 layers PCB

Table 1: *Apogee* autopilot characteristics

telemetry data that are relayed to the ground for real-time monitoring of the mission. It is also possible to store the scientific data on the autopilot SD card if the payload controller doesn't offer this functionality. An important benefit of the autopilot embedded data logger is to reduce the number of data sent to the ground compared to previous setup. The reason is that most of the bandwidth is used to send debugging information needed to investigated cases of failures after navigation errors or even crashes. With on-board data logging, all these information will be available after the flight for analysis with much more details than when using the datalink system.

http://www.imavs.org/

4 Experiments

4.1 Regulation, permit to fly

All the flights for the VOLTIGE have been performed with respect of the latest French regulation concerning MAV operation (since April 11, 2012). Two documents are needed in order to fly out of sight of the ground operator and safety pilot and at an altitude higher than 150 m: a *permit to fly* and the creation of a temporary regulated area (ZRT, Zone Reglementée Temporaire).

The *permit to fly* is a document sent to the French civil aviation authority (DGAC) that presents the aircraft, the autopilot system and all the safety procedures that will be activated in case of flying out of the zone, loss of GPS, data link or safety link,... Several categories exist with different requirements depending of the possible damages that can be caused by the aircraft. All MAV involved in the VOLTIGE project are in the smallest class category (less than 2 kg), which has relatively fewer restrictions compared to other categories.

Flying in a regulated area means that the airspace for the experiments is separated from the general airspace and no human-piloted aircraft can enter this airspace during the activated period. The request for the creation of such an area must be done several weeks in advance, which impacts the experiments since meteorological conditions are difficult to forecast more that a few days in advance. If accepted, a NOTAM is published to inform the pilots that a temporary zone has been created. The flight zone is activated the day of the flight by calling the air traffic controllers in charge of the area.

4.2 Flight campaigns

Three fields campaigns have been performed for the VOLTIGE project. The goal was to validate the capacity of the MAV for sampling the atmosphere.



Figure 4: The two types of MAV used in the VOLTIGE project

http://www.imavs.org/

First campaign The first campaign was held in Lannemezan (November 2012). The purpose of these flights was to do the initial testing and validates the choice of the airframes (Figure 4). A comparison was made between the two platforms: a glider (Multiplex Easystar) and a flying wing (Multiplex Funjet). The first plane ends up to be easier to handled, launch and land with a low flight speed, but is limited in terms of vertical climb speed and maximum wind speed for operation. The result is that the maximum altitude that can be reached is usually lower than with the other plane. For horizontal flights, it is a sane and reliable aircraft. The second flight has an higher flight speed (horizontal and vertical) with a comparable endurance. It allows to reach higher altitude (usually 50% more with similar payload).

During this first campaign, some meteorological sensors have been evaluated (temperature, pressure, humidity) by flying with two measurement units on the same aircraft (as seen Figure 4 right) and close to fixed poles (60 meters high) equipped with calibrated sensors for reference (as seen Figure 4 left).

Finally, the conditions during the campaign was pretty harsh with a lot of wind and rain. Tape have been used in order to keep the electronic dry. The selected aircraft are normal hobby planes not designed for this kind of weather. Even if the planes have satisfactory performances for the mission, it has been decide to evaluate the possibility to build a custom aircraft with an adapted payload bay if not found out of the box.

Second campaign Second campaign was at Lit-et-Mixe (French west coast) in February 2013 (Figure 5). It was the first attempt to fly within fogs since the area as the highest probability. Finally, out had no luck out of three days of flights. But other interesting meteorological phenomenon have been observed during the day with a flight almost every hours. This campaign also allowed to test all our procedures and highlight some of the limits of our system. In particular, it appears that the autonomous landing on a narrow field (short runway between trees) has to be improved in order to safely fly with reduced visibility.

Some of the preliminary scientific results are presented in section 4.3

Third campaign The third campaign was again held at Lannemezan in May 2013, with the main goal of integrating and testing a cloud sensor (Figure 6, left) under development at Reading University (UK). This was done successfully despite the difficulties to carry this extra sensor and its dedicated electronic board. From the operational point of view demonstrate once again that an integrated on-



Figure 5: Second campaign at Lit-et-Mixe. Left: sensor calibration with a reference sensor on the ground. Right: base camp of the campaign with the planes and antennas.

board electronic and an aircraft with a waterproof, easily accessible payload bay is the key point for this kind of mission.



Figure 6: Third campaign at Lannemezan. Left: base camp. Right: Easystar equipped with a cloud sensor (right wing), a video recorder (left wing) and a static electricity sensor (noise).

4.3 Preliminary scientific results

In order to validate the data measured during VOLTIGE fields campaigns, a comparison have been made with the forecasts of the AROME fine scale operational forecasting numerical model of Météo-France. Those plots (Fig.7) show the good agreement between the two types of data on some of the parameters measured.

The parameter virtual potential temperature dayly evolution (Fig. 8) permits to detect the temperature inversion level and how the boundary layer is mixed during the day. On this exemple, in the morning the boundary layer is divided in three layers. A mixing layer can be seen from the ground to 100m above it. The relative low level of humidity explains partly why fog didn't occure on this morning despite the favorable atmospheric conditions. Above this layer (around 100m a.g.l.) the strong gradient of virtual potential temperature is the marker of the inversion separating the mixing layer

http://www.imavs.org/

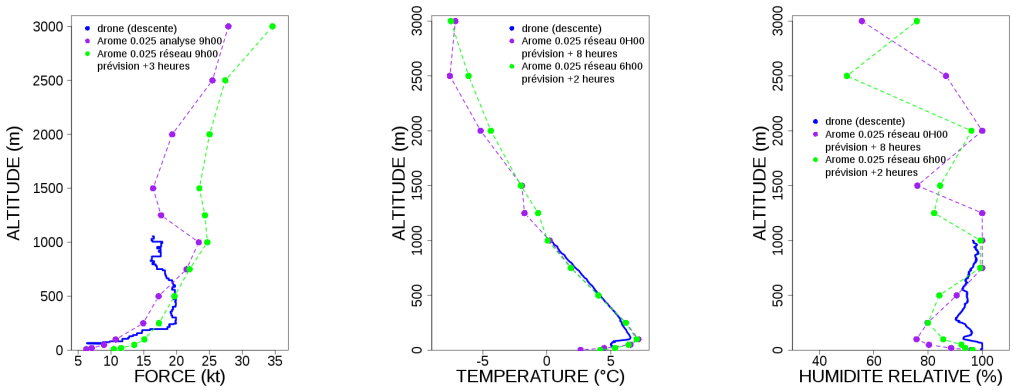


Figure 7: VOLTIGE project wind, temperature and relative humidity observations vs model forecasts

and the residual layer which is neutral with few turbulences. This is the remaining of the previous day's mixing layer. While the surface is warming during the day, the surface layer is becoming thicker and even convective (negative gradient) in the afternoon. The mixing layer is becoming thicker and overtake the residual one. At the end of the day this layer is up to 600m a.g.l. where an inversion is visible on flight n°13. This inversion seems to be the entrainment zone separating boundary layer from free atmosphere. In this case virtual potential temperature warming is probably due to the arriving of a warm front.

The measurements of the boundary layer height dayly evolution and meteorological parameters are important for the understanding of fog life cycles. The comparison to the model will be helpful to judge its ability to reproduce and forecast this phenomenon. This information will help the modelisation community to improve fog forecasting.

5 Conclusion

The VOLTIGE project is offering a good opportunity to develop new capabilities in the *Paparazzi* autopilot family, with newer processor, integrated data logger, better overall connectivity and payload management. While still at the early phase of the project, valuable experiences have been gain with three field campaigns and more than fifty flights. Among them, the constraints of operating small UAVs under bad weather conditions are leading us towards a custom payload bay with an appropriate design for easy and safe operation. The scientific results already gathered are promising, with accurate measurements and fast response time.

The next step will consists in flying multiple UAVs at the same time as a preparation for the field campaign during winter 2014 where we expect to fly in foggy conditions.

References

[1] S.T. Brown, B. Lambrigtsen, R.F. Denning, T. Gaier, P. Kangaslahti, B.H. Lim, J.M. Tanabe, and A.B. Tanner. The high-altitude mmic sounding radiometer for the global hawk unmanned aerial vehicle: Instru-

http://www.imavs.org/

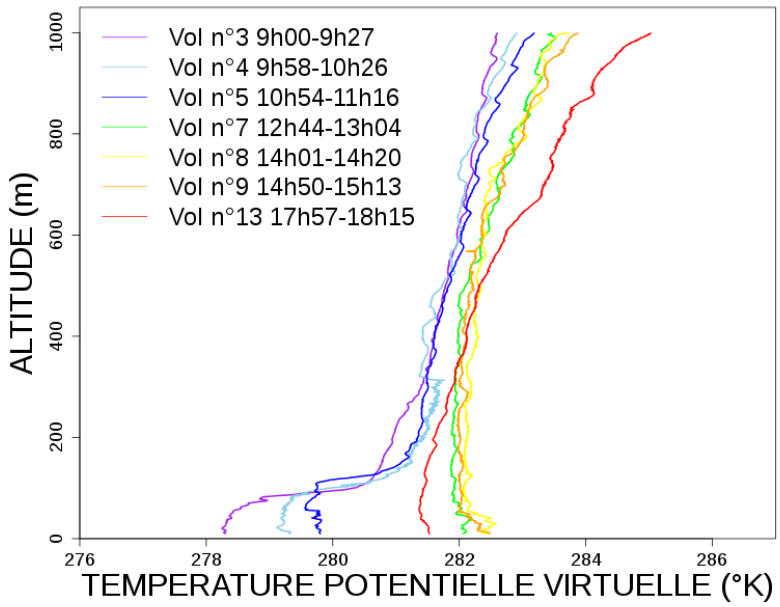


Figure 8: Virtual potential temperature evolution during the VOLTIGE project second field campaign (Lit-et-Mixe, 13/02/2013, local time).

ment description and performance. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(9):3291–3301, 2011.

[2] Hyeon-Cheol Lee. Implementation of collision avoidance system using tcas ii to uavs. *Aerospace and Electronic Systems Magazine, IEEE*, 21(7):8–13, 2006.

[3] Stephanie Mayer, Gautier Hattenberger, Pascal Brisset, Marius O. Jonassen, and Joachim Reuder. A 'no-flow-sensor' wind estimation algorithm for unmanned aerial systems. *International Journal of Micro Air Vehicles*, 4(1):15–30, 2012.

[4] J Reuder, P Brisset, M Jonassen, M Müller, and S Mayer. Sumo: A small unmanned meteorological observer for atmospheric boundary layer research. *IOP Conference Series: Earth and Environmental Science*, 1(1):012014, 2008.

[5] K.J. Rogers and A. Finn. Frequency estimation for 3d atmospheric tomography using unmanned aerial vehicles. In *Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on*, pages 390–395, 2013.

[6] F. Wantuch, Z. Bottyan, Z. Tuba, and K. Hadobacs. Statistical methods and weather based decision making in meteorological support for unmanned aerial vehicles (uavs). In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pages 203–207, 2013.

http://www.imavs.org/

Technical Session 3

Aerodynamics

Chairman : Emmanuel Benard

- Sergey Serokhvostov, Kirill Shilov and Nikita Pushchin. ***MAV unsteady characteristics in-flight measurement with the help of SmartAP autopilot***
- Warakorn Hlusriyakul, Chanchai Pattanathummasid, Chinnapat Thipyopas and Chanin Tongchitpakdee. ***Aerodynamic Investigation and Analysis of Wingtip Thickness's Effect on Low Aspect Ratio Wing***
- Kwanchai Chinwicharnam, David Gomez Ariza, Jean-Marc Moschetta and Chinnapat Thipyopas. ***Aerodynamic Characteristics of a Low Aspect Ratio Wing and Propeller Interaction for a Tilt-Body MAV***
- C.-H. Kuo, C.-M. Kuo, A. Leber and C. Boller. ***Vector thrust multi-rotor copter and its application for building inspection***

http://www.imavs.org/

MAV Unsteady Characteristics in-Flight Measurement with the Help of SmartAP Autopilot

S. Serokhvostov, N. Pushchin and K. Shilov

Moscow Institute of Physics and Technology
Department of Aeromechanics and Flight Engineering
140180, Gagarina street, 16, Zhukovsky, Russia
serokhvostov@phystech.edu

Abstract

Proposed is the method of measuring the unsteady characteristics of Micro Air Vehicles (MAV) in flight with the help of small autopilot. Theoretical background, experimental methodic and results obtained are presented. Comparison with theoretical results is made.

http://www.imavs.org/

1 Introduction

Unsteady characteristics of micro air vehicles (MAVs) can strongly affect on its flight performance (stability, controllability etc.). But up to now there was only the investigations of such characteristics for rather large airplanes. The characteristics for small ones can differ from those because of rather small values of Reynolds numbers and aircraft inertia characteristics. But the knowledge of these characteristics is important for the MAV design. So, it is required to check if the formulas for large aircrafts are applicable for small ones or we need to correct these formulas.

Some difficulties occur for the measurement of these characteristics in the wind tunnel. There must be special devices for model movements and rather precision measurement devices.

The another method of determination can be the flight test. To obtain the necessary data one must measure kinematical flight parameters (speed, rotation frequency, orientation of aircraft) and derive the unsteady characteristics.

One of the ideas of this investigation is to use autopilot system as a flight parameters recorder. The second idea is to separate the influence of inertia parameters (mass, inertia momentum) and the influence of aerodynamical effects. This can be made by changing the mass and changing the mass distribution in the aircraft. Using this technology one can analyze the influence of mass and center of gravity position, inertia momentums on the flight characteristics.

2 Experimental Equipment

2.1 Registration device

The main idea for the flight parameters registration is to use the small-sized autopilot. Indeed, the autopilot has all the sensors required and have many other useful features. Among the autopilots available the SmartAP autopilot [3], [1] was chosen, see fig. 1. The main features of this device are:

- light weight, it enables to use this device in rather small air vehicles;
- powerful microcontroller, which can give an opportunity to run routines at high frequencies;

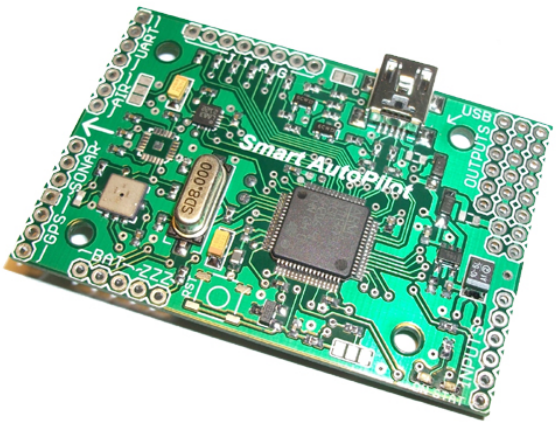


Figure 1: SmartAP autopilot

- 9-Degrees Of Freedom inertial measurement unit, which includes 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer in order to determine attitude;
- pressure sensor for altitude measurement;
- differential pressure sensor for air speed measurement;
- GPS receiver for global position determination;
- wireless data channel for two-way telemetry;
- SD-card for in-flight data logging;
- USB port for PC/Laptop connection for firmware uploading, debugging and testing routines;
- PWM inputs/outputs to read signals from standard RC receiver and send them to motors/servos;
- Several ADC channels for battery voltage/amperage monitoring.

On the first stage it was decided to control the aircraft manually and collect the flight data and control signals on SD-card onboard. But it should be mentioned that this autopilot enables to make the measurements in fully automatic mode, i.e. the autopilot can guide the aircraft along the flight path, then make some disturbance, write the aircraft reaction to the SD-card and then correct the flight path.

Therefore, SmartAP autopilot is installed in-between standard RC receiver and servos. It provides the capability to register both pilot's input data and aircraft's behavior in form of linear accelerations, angular rates, Euler angles and airspeed measured by differential pressure sensor (Pitot tube). The logging frequency chosen is 50 Hz. Acceleration's measurement range was set to ± 4 g, gyroscopes range was set to ± 2000 deg/s.

2.2 Aircraft

As an experimental airplane the "Progress" model aircraft [4] was chosen (see fig. 2). The main characteristics of this plane are: mass — 150 gram; wingspan — 85 sm; wing aspect ratio — 3.9; wing

http://www.imavs.org/

http://www.imavs.org/



Figure 2: "Progress" model airplane with autopilot

taper ratio — 2.16; wing area — 0.82 m^2; horizontal tail surface — 0.053 m^2; horizontal tail distance from CG — 0.47 m; vertical tail surface — 0.0428 m^2; vertical tail distance from CG — 0.47 m.

This aircraft has rather low mass for this wing area, so the inertial effects in unsteady motion (including rotational ones) are much more less than the damping aerodynamic effects. This enables to investigate more easy the unsteady aerodynamical derivatives.

3 Theoretical Background

3.1 Conceptual Questions

The equations of disturbed motion form the system of 8 differential equations [2] that is very difficult or impossible for analytical research. One of the ideas for the analysis simplification was [2] to divide the motion on longitudinal and lateral.

The assumptions for the separation of the disturbed motion on lateral and transversal are [2]:

1. disturbances are small enough,

2. aircraft flies along the straight line without acceleration, without bank and slip,

3. air density is constant (flight altitude change is small enough),

4. aircraft is symmetrical along the vertical plane.

Also it follows from above assumptions that the aircraft must fly at the regimes without flow separation (as separation makes flow asymmetrical).

No more assumptions about the aircraft size and design and the Reynolds and Stroughal numbers and other parameters of similarity have not been made. So, as a conclusion, one can analyze the lateral and transversal motions of MAVs separately at least at rather important case of straight flight without acceleration and without flow separation.

So, at this stage of investigation the flight regimes were limited by the case of level flight with constant velocity.

http://www.imavs.org/

3.2 Longitudinal Motion

Longitudinal disturbed motion (in Russian tradition) is usually analyzed in the coordinate system where OX axis is collinear to the velocity vector of aircraft, OY axis is perpendicular to OX and lies in the plane of aircraft symmetry and OZ axis is along the right semi-wing.

The values of characteristic scale of time

$$\tau = \frac{2m}{\rho S V_0},$$

and dimensionless aircraft density

$$\mu = \frac{2m}{\rho S L},$$

are required for further investigation (m is mass of MAV, ρ — air density, S — wing area, V_0 — non-disturbed flight velocity, L — wing span).

Let's make the following assumptions.

1. Mach number M is zero and all the coefficient's derivatives with respect to M are equal to zero.

2. Thrust vector is directed along the OX axis and is passed through the aircraft center of gravity.

3. Flight altitude is constant.

Under these assumptions the disturbed motion is described by the system of four non-dimensional linearized equations [2]. The solution of this system is in the form of $A\exp(\lambda t/\tau)$, where A is defined by initial conditions and λ is obtained from characteristical equation

$$\lambda^4 + a_1\lambda^3 + a_2\lambda^2 + a_3\lambda + a_4 = 0.$$

In common case λ is complex value. If the real part of λ is negative, then the disturbed motion is damped. This case is required for the stability conditions.

Coefficients of the characteristical equation can be found from formulas [2]

$$\begin{cases} a_1 &= C_L^\alpha + 3C_D - \dfrac{m_z^{\omega z} + m_z^{\dot\alpha}}{r_z^2} - \dfrac{2T^V V_0}{\rho V_0 S}, \\ a_2 &= -\dfrac{\mu m_z^\alpha + C_L^\alpha m_z^{\omega z}}{r_z^2} - \dfrac{m_z^{\omega z} + m_z^\alpha}{r_z^2}\left(2C_D - \dfrac{2T^V V_0}{\rho V_0^2 S}\right) + \\ & \quad +2\left(C_L^2 + (C_L^\alpha + C_D)\left(C_D - \dfrac{T^V V_0}{\rho V_0^2 S}\right) - C_L C_D^\alpha\right), \\ a_3 &= -2\left(C_L^2 + (C_L^\alpha + C_D)\left(C_D - \dfrac{T^V V_0}{\rho V_0^2 S}\right) - C_L C_D^\alpha\right)\dfrac{m_z^{\omega z}}{r_z^2} - \\ & \quad -2C_L^2\dfrac{m_z^{\dot\alpha}}{r_z^2} - 2\left(C_D - \dfrac{T^V V_0}{\rho V_0^2 S}\right)\dfrac{\mu m_z^\alpha}{r_z^2}, \\ a_4 &= -2C_L^2\dfrac{m_z^\alpha}{r_z^2}, \end{cases}$$

where $r_z^2 = I_{ZZ}/(mb^2)$ is dimensionless MAV radius of inertia for OZ axis, I_{ZZ} — aircraft moment of inertia about an axis OZ, b — mean wing chord, C_L — lift force coefficient, C_D — drag force coefficient, C_i^j — force coefficient derivatives with respect to some variable j, m_z^j — moment coefficient derivatives with respect to some variable j, α — angle of attack, ω_z — rotational frequency about an axis OZ, T^V — thrust derivative with respect to V.

The criteria for the stability of motion are
$a1 > 0$, $a2 > 0$, $a3 > 0$, $a4 > 0$, $R = a_1a_2a_3 - a_1^2a_4 + a_3^2 > 0$.

The requirement $a_4 > 0$ assumes that $m_z^\alpha < 0$. This condition can be easily achieved by proper position of the center of gravity (forward of the aerodynamic center (AC)).

For the electrical powerplant $2T^V/(\rho V_0 S)$ is negative and of order of $(-C_D)$. So, the condition $a_1 > 0$ is certainly valid if $m_z^{\dot\alpha} < 0$ (for the most of cases $m_z^{\omega z} < 0$).

With the help of formulas from [2] it was found that $C_L \approx 0.5$, $C_D \approx 0.06$, $C_{D0} < 0.03$, $m_z^{\omega z} + m_z^{\dot\alpha} \approx \approx -5.7$ for horizontal tail, $m_z^{\omega z} \approx 0.3$ for the wing, $C_L \approx 3.6$, $\mu \approx 1.7$, $r_z \approx 0.7$, $m_z^\alpha \approx -0.18$, $\mu = 1.7$. For these values the coefficients a_2 and a_3 are positive. Also the above assumptions can help to simplify the expressions for a_1, a_2, a_3:

$$a_1 \approx C_L^\alpha - \frac{m_z^{\omega z} + m_z^{\dot\alpha}}{r_z^2},$$
$$a_2 \approx -\frac{\mu m_z^\alpha + C_L^\alpha m_z^{\omega z}}{r_z^2},$$
$$a_3 \approx -2C_L^2\frac{m_z^{\dot\alpha}}{r_z^2} - 2C_D\frac{\mu m_z^\alpha}{r_z^2}.$$

Now the requirement of $a_2 > 0$ is well known condition

$$m_z^{C_L} + \frac{m_z^{\omega z}}{\mu} > 0.$$

For the aircraft investigated the second term is not as low as for the larger aircraft. For $\mu = 1.7$ it is much more higher than the first summand.

In expression for a_1 the first term occurs smaller than the second. From this, one can see that with the changing of the r_z all coefficients a_i change nearly proportionally, so one can obtain rather little information changing r_z.

http://www.imavs.org/

The next question is the ability to separate the disturbed motion onto short-periodic and long-periodic. One of the ways of roots obtaining described in [2] assumes that there are two pairs of complex conjugated roots, and one pair is much enough comparing to the other with respect to the imagine parts (oscillation frequencies). So, one can obtain the larger roots (in absolute value) from the equation

$$\lambda^2 + a_1\lambda + a_2 = 0$$

and the smaller roots from equation

$$a_2\lambda^2 + a_3\lambda + a_4 = 0.$$

If these roots are really differ significantly, so it is possible to investigate short-period and long-period disturbance separately. The solution of second order equation is well known, so

$$\lambda_{1,2} = \frac{-a_1 \pm i\sqrt{4a_2 - a_1}}{2}, \quad \lambda_{3,4} = \frac{-a_3 \pm i\sqrt{4a_2a_4 - a_3^2}}{2a_2}.$$

3.3 Lateral Motion

Usually [2] the lateral motion is analyzed in the coordinate system in which OX axis is oriented along the wing chord of airplane, OY axis is perpendicular to OX and lies in the symmetry plane of aircraft and OZ axis is perpendicular to OX and OY (and oriented along the wing length).

The necessary relationships required for lateral motion analysis in the coordinate system defined above are represented below.

If β is sideslip angle, then for the wing moment characteristics are [2]

$$\begin{aligned} {m_x}^{\beta}{}_W &\approx -0.5C_L^{\alpha}\bar{z}\cos^2(\chi)\psi - 0.5\bar{z}\sin(\chi)C_L, \\ {m_y}^{\beta}{}_W &= -0.06C_L^2\mathrm{tg}(\chi) - \alpha {m_x}^{\beta}{}_W, \end{aligned}$$

where $\bar{z}$ is the distance from OX axis to the "semi-wing center of gravity" with respect to semi-wingspan, $0.33 < \bar{z} < 0.5$, χ — wing sweep angle, ψ — wing dihedral.

Torque coefficients due to the rotation for the wing are [2]

$$\begin{aligned} {m_x}^{\omega x}{}_W &= -\frac{\xi}{2}\left(2C_L\alpha + k_iC_L^{\alpha}\right), \\ {m_x}^{\omega y}{}_W &= -\frac{\xi}{2}(2 - k_i)C_L, \\ {m_y}^{\omega x}{}_W &= -\xi\left(\left(C_D - C_L\alpha\right)\alpha + C_L\left(\frac{C_L^{\alpha}}{\pi\lambda} - 1\right)k_i\right), \\ {m_y}^{\omega y}{}_W &= -\xi\left(\left(C_D - C_L\alpha\right) - C_L\left(\frac{C_L^{\alpha}}{\pi\lambda} - 1\right)\alpha k_i\right), \\ k_i &= 0.9(0.5 + 0.033\lambda), \end{aligned}$$

where ξ is inertia radius in second power of semi-wing from the symmetry axis with respect to semi-wingspan, $0.33 < \xi < 0.5$.

http://www.imavs.org/

For the vertical tail

$$
\begin{aligned}
{m_x}^{\beta}{}_T &= k\frac{S_T y_T}{SL}{C_Z}^{\beta}{}_T,\\
{m_y}^{\beta}{}_T &= k\frac{S_T L_T}{SL}{C_Z}^{\beta}{}_T,\\
{m_x}^{\omega x}{}_T &= 2k\frac{S_T}{S}\left(\frac{y_T}{L}\right)^2{C_Z}^{\beta}{}_T,\\
{m_y}^{\omega x}{}_T &= 2k\frac{S_T}{S}\frac{y_T}{L}\frac{L_T}{L}{C_Z}^{\beta}{}_T,\\
{m_x}^{\omega y}{}_T &= 2k\frac{S_T}{S}\frac{y_T}{L}\frac{L_T}{L}{C_Z}^{\beta}{}_T,\\
{m_y}^{\omega y}{}_T &= 2k\frac{S_T}{S}\left(\frac{L_T}{L}\right)^2{C_Z}^{\beta}{}_T,\\
C_Z{}^{\beta}_T &< 0
\end{aligned}
$$

where S_T — vertical tail area, y_T — location of vertical tail aerodynamic center along the axis OY, L_T — location of vertical tail aerodynamic center from the center of gravity, k — flow decrease coefficients on the vertical tail.

For further investigations the characteristic "lateral" scale of time

$$\tau_\perp = \frac{m}{\rho S V_0}$$

is required.

Dimensionless radii of inertia are

$$r_{XX} = \frac{4I_{XX}}{mL^2}, \quad r_{YY} = \frac{4I_{YY}}{mL^2}$$

where I_{XX} and I_{YY} are inertia moments of airplane about the axes OX and OY. It is assumed that $I_{XY} = 0$.

All torque coefficients below are normalized (assumed to be divided on the appropriate dimensionless inertia radii):

$$\bar{m}_i^q = \frac{m_i^q}{r_{ii}}.$$

With the help of the above relationships the characteristical values λ of dimensionless disturbed motion for level flight can be found from the characteristic equation [2]

$$\lambda^4 + a_1\lambda^3 + a_2\lambda^2 + a_3\lambda + a_4 = 0$$

where [2]

$$
\begin{aligned}
a_1 &= -\left(0.5C_Z^{\beta} + \bar{m}_x^{\omega x} + \bar{m}_y^{\omega y}\right),\\
a_2 &= 0.5C_Z^{\beta}(\bar{m}_x^{\omega x} + \bar{m}_y^{\omega y}) + (\bar{m}_x^{\omega x}\bar{m}_y^{\omega y} - \bar{m}_x^{\omega y}\bar{m}_y^{\omega x}) - \mu(\bar{m}_y^{\beta} + \bar{m}_x^{\beta}),\\
a_3 &= -(0.5C_Z^{\beta}(\bar{m}_x^{\omega x}\bar{m}_y^{\omega y} - \bar{m}_x^{\omega y}\bar{m}_y^{\omega x}) + \mu(\bar{m}_x^{\beta}\bar{m}_y^{\omega x} - \bar{m}_y^{\beta}\bar{m}_x^{\omega x}) -\\
&\quad - \mu\alpha(\bar{m}_x^{\beta}\bar{m}_y^{\omega y} - \bar{m}_y^{\beta}\bar{m}_x^{\omega y}) + 0.5\mu C_L\bar{m}_x^{\beta}),\\
a_4 &= 0.5\mu C_L(\bar{m}_x^{\beta}\bar{m}_y^{\omega y} - \bar{m}_y^{\beta}\bar{m}_x^{\omega y}),\\
C_Z^{\beta} &= {C_Z}^{\beta}{}_T S_T/S_W.
\end{aligned}
$$

http://www.imavs.org/

For the most part of cases two roots of the characeristical equation are real and two others are complex conjugate values (these roots are responsible for so-called "Dutch step").

Preliminary calculations show that not all the coefficients a_i change proportionally to each other with radii of inertia change. So, changing the inertia radii one can separate the influence of aerodynamical effects and inertia effects.

4 Theoretical and Experimental Results

Above formulas enable to obtain the values of λ_i from the characteristic equations for longitudinal and lateral motion. Also it is possible to obtain these values from experimental data. So, one can compare the reaults and make the conclusion about the applicability the well-known formulas to the case of low Reynolds numbers.

On the basis of data for the model airplane PROGRESS the inertia moments and dimensionless coefficients for this aircraft were obtained.

For the longitudinal motion the following coefficients were obtained: $a_1 = 14.4$, $a_2 = 23.1$, $a_3 = 2$, $a_4 = 0.0037$.

From these data one can obtain that all the roots of characteristic equation are real and all of them are negative. It means that the aircraft is stable in lateral motion. The values of roots are $\lambda_1 = -12.53$, $\lambda_2 = -1.84$, $\lambda_3 = -0.085$, $\lambda_4 = -0.0019$.

Characteristical time is $\tau \approx 0.47$ seconds.

First of all, one can see that the roots form two pairs. One pair are the roots with high values, the other is pair with small values. So, at least for this case it is possible to separate the disturbed motion onto "short-periodic" and "long-periodic".

The absence of "oscillations" (imagine part of the roots) can be explained by very high damping moment of horizontal tail.

To obtain the experimental data required it is necessary to have a part of flight without control impacts.

Figure 3 shows the record of flight data (pitching angle vs time). There is rather long path with no influence of elevator.

One can see that there is practically no oscillations of disturbed motion (a "little wavy" shape can be due to the atmosphere disturbances) as it was predicted by theoretical results.

The same situation is for lateral motion (see fig. 4).

A set of preliminary experiments was also made with changing the inertia radius. For this purpose two additional masses were placed on the wing tips (you can see these masses in fig. 2).

5 Present situation, future work and lessons learned

For this moment the tuning of autopilot was made that enable to record flight parameters. A set of test flights were made on the experimental airplane with autopilot and the records of flight parameters were obtained.

Theoretical results qualitatively coincide with flight data in first approximation. But for the increasing the accuracy of experiment some additional steps must be made. For example, it can be necessary to make the mode of flight with recording "more automatic".

Also, it can be required to increase the accuracy of theoretical data calculations.

http://www.imavs.org/

http://www.imavs.org/

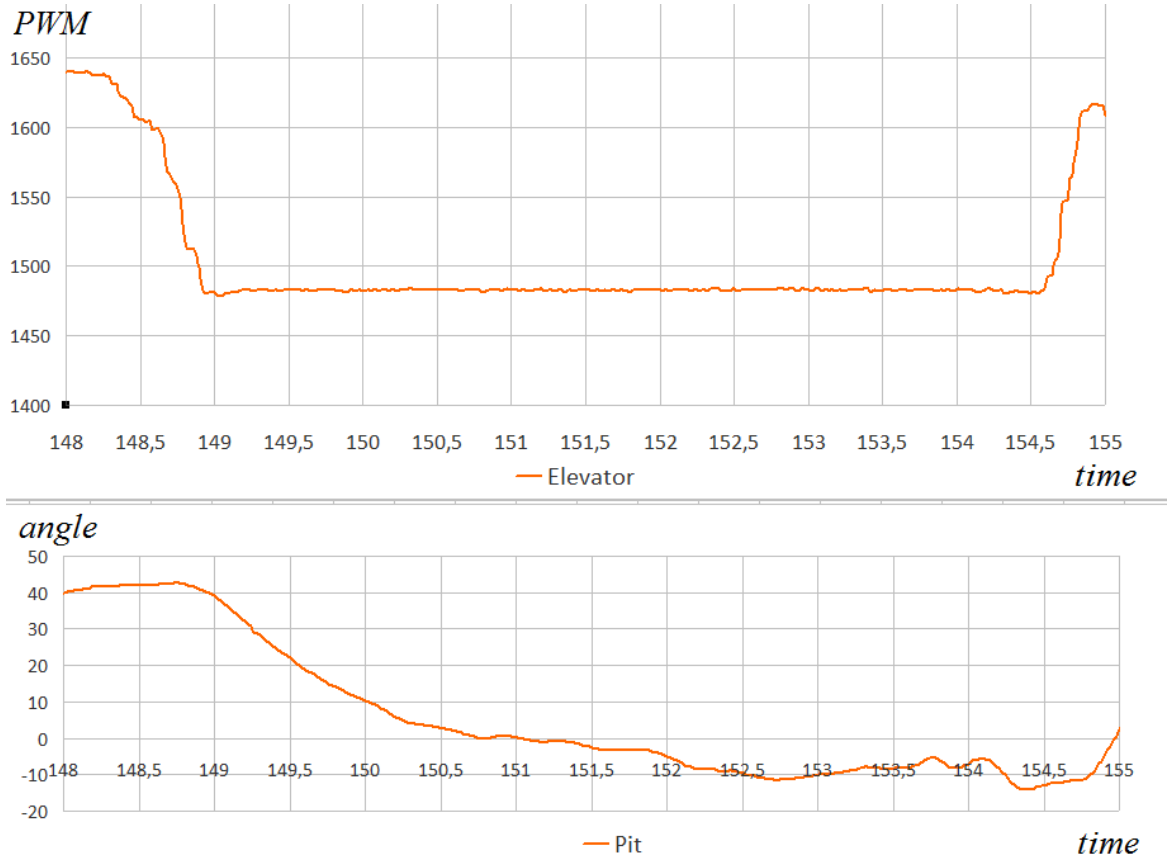


Figure 3: Control signal on elevator (up) and pitch angle (bottom) vs time

Of cause, both experimental and theoretical works must be continued. One more task that must be investigated is the behavior of aircraft at high angles of attack and the steady and unsteady characteristics.

Also the other types of aircrafts (tailless configuration and unconventional configurations) is planned to be investigated.

6 Conclusion

First stage of unsteady MAV characteristics obtaining methodic development with the help of micro autopilot was made. Theoretical data were obtained and a set of flight experiments were made.

Transition from manual to automatic aircraft control is the next step in presented research. This assumes increased accuracy of induced disturbances.

References

[1] Smartap autopilot. [online]. Available at `http://sky-drones.com/`.

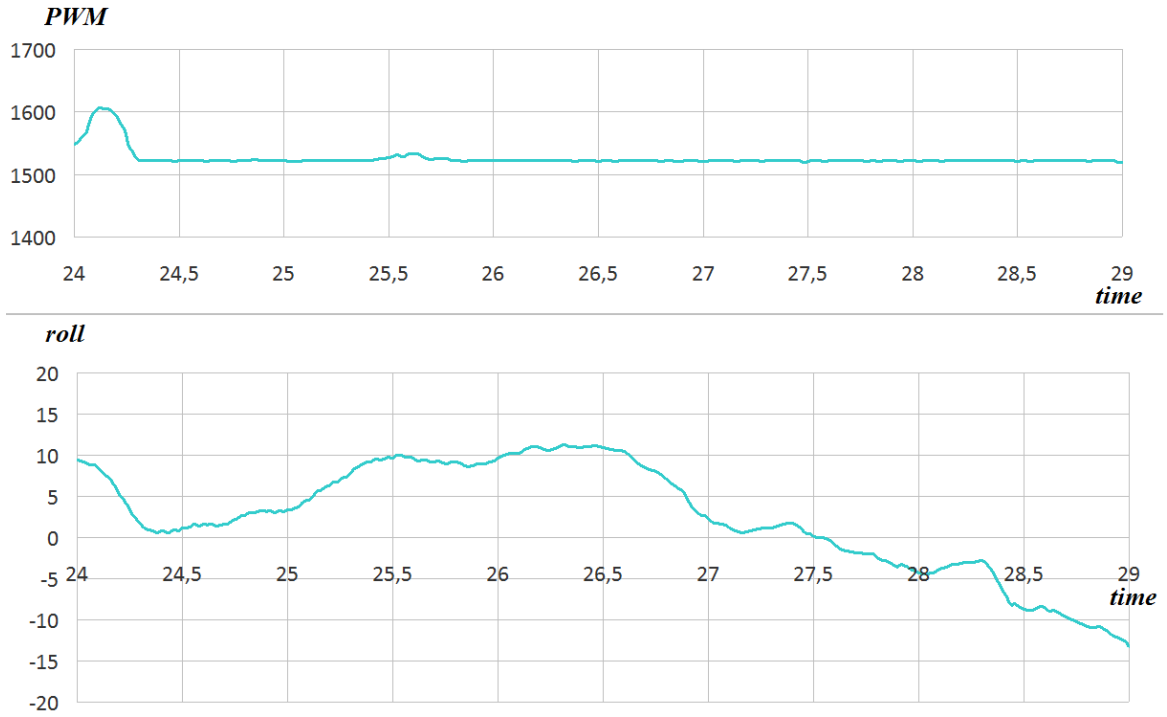


Figure 4: Control signal on aileron (up) and roll angle (bottom) vs time

[2] I. V. Ostoslavsky. *Aircraft Aerodynamics (in Russian)*. Oborongiz, Moscow, 1957.

[3] Kirill Shilov and Grigory Lazurin. MAV autopilot for commercial and research purposes. In *Proceedings of the Conference IMAV 2012*, Braunschweig, Germany, 3-6 July 2012.

[4] Guy Varneuil. Progress sans stress! *Fly International*, (202), January 2012.

http://www.imavs.org/

Aerodynamic Investigation and Analysis of Wingtip Thickness's Effect on Low Aspect Ratio Wing

Chanin Tongchitpakdee*, Warakorn Hlusriyakul†, Chanchai Pattanathummasid‡ and Chinnapat Thipyopas§

Department of Aerospace Engineering, Kasetsart University, Bangkok, Thailand
chanin.t@ku.ac.th

http://www.imavs.org/

Abstract

In this present work, the tailless fixed-wing MAV, KUMAV-001, which is an in-house built low-aspect-ratio monoplane with span of 45 cm, aspect ratio of 1.07, and Eppler-212 airfoil section, will be used as a study case. The Eppler-212 airfoil has thickness about 10% of chord and it provides enough space at the central part. In this paper, the aerodynamics of the KUMAV-001 MAV operating at Re of 250,000 is studied. The effects of the maximum thickness variation of the Eppler -212 airfoil at wing tip are investigated. Analyses of the overall wing aerodynamic performance are explored using computational fluid dynamic (CFD) commercial software and wind tunnel tests. All longitudinal aerodynamic characteristics will be computed and measured. Referenced experimental data are used to validate the CFD simulation. Results indicate that, at low angle of attack condition, the aerodynamic performance (lift-to-drag ratio) can be improved by thin wingtip design. At high angle of attack condition, however, implementing the thin wingtip design does not appear to be effective in increasing lift-to-drag ratio.

1 Introduction

Micro Air Vehicle (MAV) is a small flying vehicle which usually flies in low Reynolds number regime. Due to the constraint of size, most fixed-wing MAVs are commonly designed with low- aspect-ratio tailless wing configuration. Fuselage is generally combined to host all equipment and payloads. Though the low-aspect-ratio wing provides very high stall angle of attack (AOA), it introduces high induced drag due to the effect of tip vortices. Furthermore, aerodynamic performance of wings in low Reynolds number regime is typically low because of unexpected flow separation. An interference drag between wing and body also plays an important effect. Accordingly, these result in a reduction of MAV's aerodynamic performance. Therefore, one of the challenges for aerodynamic and design of MAV is to improve lift-to-drag ratio (L/D) of overall vehicle. Generally, thin airfoils give better aerodynamic performance [1] and hence suitable for MAV design. However, using of the thin wing design has a drawback of limited space required for housing all equipment and payloads. The fuselage must be added leading to an increment of the interference drag. Thus, an idea of utilizing thick airfoil at root for installing components while decreasing the wing thickness on the outboard part to improve aerodynamic performance of the wing is proposed.

*Lecturer in Aerodynamics and Aircraft Design
†Undergraduate student (north_wimer@hotmail.com)
‡Undergraduate student (chanchai.p@hotmail.com)
§Lecturer in Aerodynamics and Aircraft Design (fengcpt@ku.ac.th)



Figure 1: KUMAV-001 Design

Airfoil	E212	Max. Thickness	10.55
Root chord	45 cm	Tip chord	39 cm
Aspect ratio	1.07	Span	45 cm
LE swept	17 deg	Dihedral	0 deg
Control surface	20% root chord	Flight speed	7-15 m/s

Table 1: Design configuration of KUMAV-001

MAV activity in Kasetsart University (KU) has started since 2010. The first MAV, named KUMAV-001, has been constructed and tested. KUMAV-001 design consists of propulsive tractor with elevon configuration. It has 45cm-span, aspect ratio of 1.07 and Eppler-212 airfoil section. The baseline wing model came from the monoplane wing studied by Arizona State University (ASU) [2]. Fuselage is added in order to install battery and other components. A year later, the effects of winglet on MAV performance have been investigated by Thipyopas and Intaratep [3]. Wind tunnel and flight test were conducted to study the propwash effect to the wing. The design and study of KUMAV-001 was presented in the IMAV2011 conference. The KUMAV-001 is illustrated in Figure 1 and summary data of design configuration are shown in Table 1. The lift-to-drag ratio (L/D) of KUMAV-001, however, is quite low like other low-aspect-ratio monoplane MAVs. To improve flight endurance, KUMAV-001 was redesigned to achieve higher L/D. Mueller [4] conducted experimental tests to compare thick and thin airfoils at low Reynolds numbers for MAVs. He found that thin airfoils give greater aerodynamic performance. Viieru et al. [5] enhanced the L/D of MAV by modification of wingtip shape. In 2008, Blanc et al. [6] studied various thickness of wing using NACA 53xx airfoil sections. The wing thickness is varied from 2% to 10% chord. The wing thickness of 2% chord gave the best result of the L/D. This suggested that thin airfoil wing is suitable for MAV design. However, reducing the wing thickness too much may result in decreasing L/D. A drawback of thin wing design is that it has no enough space for installing battery and equipment. The fuselage must be then added to accompany all the payloads. Adding the fuselage may cause the interference drag. Thus, an idea of remaining thick airfoil at wing root in order to install components while decreasing of thickness on outboard part, where no need to attach any component, is studied in this paper. Since the thickness of Eppler-212 airfoil is about 10% chord, it provides enough space at the central part. This concept may be beneficial and practical for low-aspect-ratio MAVs.

http://www.imavs.org/

2 Methodology

This section provides the descriptions of apparatus and procedures of the wind tunnel tests and CFD simulations studied in the present work.

2.1 Wind Tunnel Tests

Five wing-only models were designed based on the design configuration as shown in Table 1. Effects of engine, fuselage, and control surfaces are neglected. The baseline wing model has a constant thickness (10.55% chord), while other four models had thickness varying from 10.55% chord at the root to approximately 8%, 6%, 4% and 2% chord at the tip, respectively. All tested models, illustrated in Figure 2, were made of foam covered by composite material for rigidity, and wrapped by thin plastic sheet for smooth surface. Figure 3 shows schematics and dimensions of the geometries for the wing model with thickness of 2% chord at the tip.

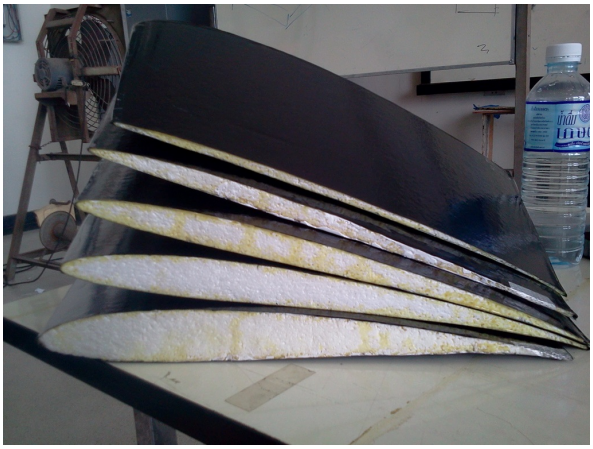


Figure 2: Test models

The aerodynamic characteristics were measured by a 3-component balance. The wing model was supported by a single strut at the mean chord position. Lift, drag, and pitching moment were obtained directly by a 3-component balance system outside the test section. The maximum capacities are ± 15 N and ± 1.5 N-m for force and moment, respectively. The accuracies of the balance for force and moment are 0.1 N and 0.02 N-m, respectively. The balance and angle of attack adjustment systems using the turn table and test setup are shown in Figure 4. All experiments were conducted in the closed-loop wind tunnel at Kasetsart University. The wind tunnel has a test section of 1m $\times$ 1m $\times$ 3m (W$\times$ H $\times$L), and a contraction ratio of 4. The maximum speed is 60 m/s generated by a 2m-diameter fan with maximum power consumption of 75 kW. The test speed presented in this study is 10 m/s only. Test speed was measured using a Pitot-static probe installed in front of the model and a digital manometer with an accuracy of 5 Pa. For the wind tunnel testing, lift, drag, and pitching moment were measured with angle of attack ranged between -9 and 36 degrees. Data correction was performed for the wind tunnel test data. The standard blockage and wall correction of Barlow et al. [7] was applied to the primary measured data. The schematics of the wind tunnel are shown in Figure 5.

http://www.imavs.org/

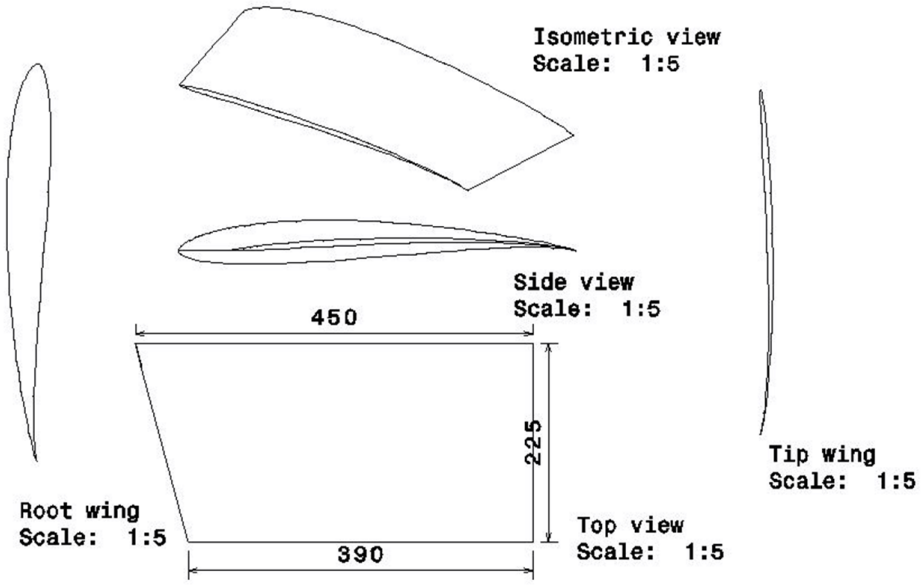


Figure 3: Geometry for a wing model with tip thickness of 2% chord



Figure 4: 3-component balance with turn table and model setup

http://www.imavs.org/



Figure 5: Schematic views of Kasetsart University wind tunnel

http://www.imavs.org/

2.2 CFD Simulation

In this present study, all CFD simulations were performed with a CFD commercial solver FLUENT^{TM} 6.3. Flow fields were simulated and analyzed with three-dimensional steady incompressible Reynolds-Averaged Navier-Stokes (RANS) simulations at Reynolds number of 250,000. To simulate the effect of turbulence, the Spalart-Allmaras one-equation turbulence model was used. Five different wing geometries were modeled and the computational grids were generated with structured mesh as shown in Figure 6. The grid is clustered in the vicinity of the wing surfaces, the trailing edge, and the wing tip to accurately capture the effects of tip vortices and trailing edge wakes. The far field boundaries are placed approximately 10 chords upstream and 20 chords downstream the wing. The far field boundary in spanwise direction is located approximately 10 half-span of the wing away from the tip.

On the upstream and far field boundaries, a velocity inlet condition is imposed with axial velocity Vx = 10 m/s. On the downstream boundary, a pressure outlet condition is imposed with pressure of 101,325 Pa. Wall conditions (no-slip conditions) are imposed on the wing surfaces. A symmetry condition is imposed on the mid-span plane to simulate the effect of another half wing. The reference values for these simulations are: freestream temperature T_∞ = 288.16 K, freestream density ρ_∞ = 1.225 kg/m^3, reference chord length c = 0.42 m, and reference wing area = 0.0945 m^2. Validation of CFD simulation was performed. The grid-independence study yields a reasonable good mesh of 4.28 $\times$ 10^6 grid points to be used in further simulations.

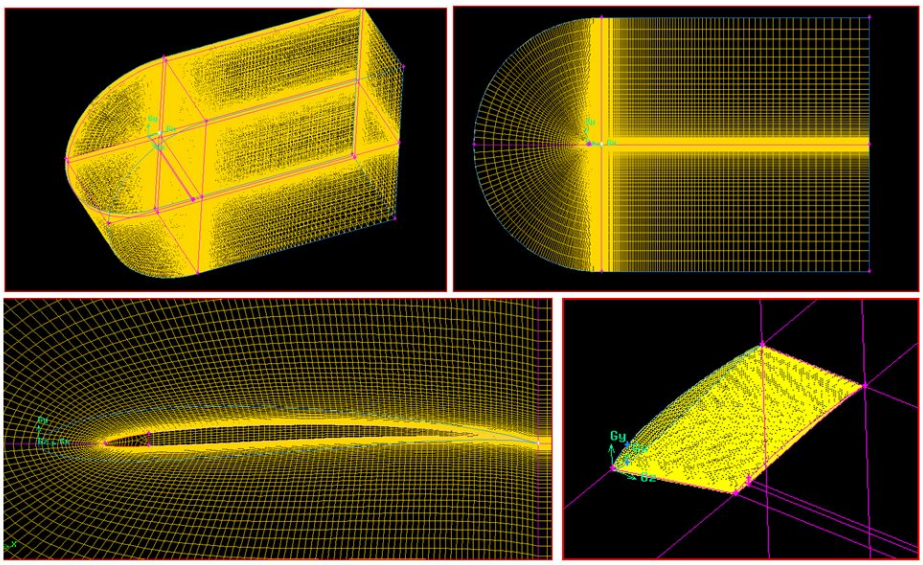


Figure 6: 3D structured grid for CFD simulations

3 Results and Discussion

The following wind tunnel and CFD results have been obtained for the several wing configurations at a flight speed of 10 m/s (Re = 250,000) and at angles of attack ranged from -10 (from 0 in CFD) to 35 degrees. The effect of wingtip thickness on wing aerodynamic performances was studied by comparing the predicted lift coefficient (C_L), drag coefficient (C_D), pitching moment coefficient (C_M), and lift-to-drag ratio (L/D) of the modified wings with the baseline wing. For the following results, the numbers showed in the plots refer to the wingtip thickness in percentage (with 2 decimals) of root chord length; for example, '1055' means 10.55%c thickness, '396' means 3.96%c thickness

3.1 Validation of CFD Solver

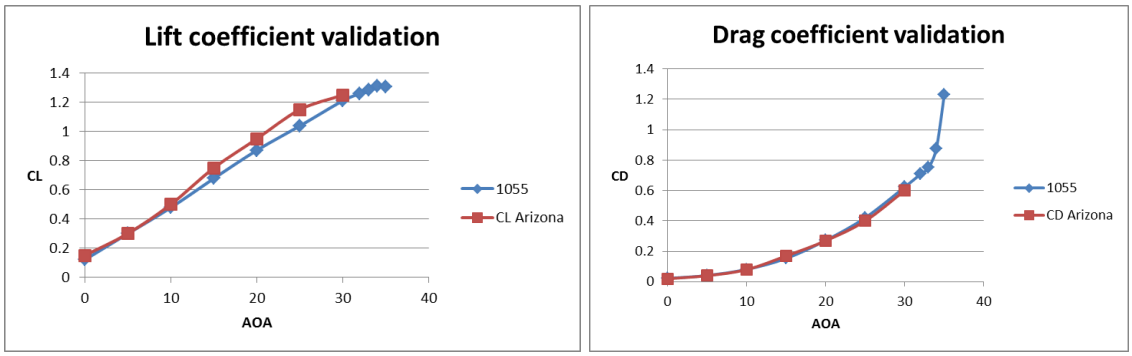
Prior its further use to investigate wing models with various thicknesses, the flow solver has been validated for the baseline wing (10.55%c thickness) with reference data from the study of Arizona State University (ASU) [2]. To validate the solver, plots of lift coefficient and drag coefficient varied with angle are given in Figure 7. It is clearly seen that the validation gives a reasonable good agreement between CFD results and experiments.

3.2 Effects of Wingtip Thickness on Aerodynamic Performances

Figure 8 shows the variations of lift coefficient as a function of angle of attack (AOA). Experiments and simulations clearly show that the thickness of wingtip has strong effect on aerodynamic performance of the wing, particularly at high AOAs. For the thinner wingtip, the lift coefficients are higher relative to the baseline constant-thickness wing. It is also seen that lift curve slope increases as wingtip thickness decreases. The wind tunnel results appear to have some non-linearity in the lift curve slope possibly due to some uncertainty in the angle of attack. The wings with thinner tip have lower stall angle of

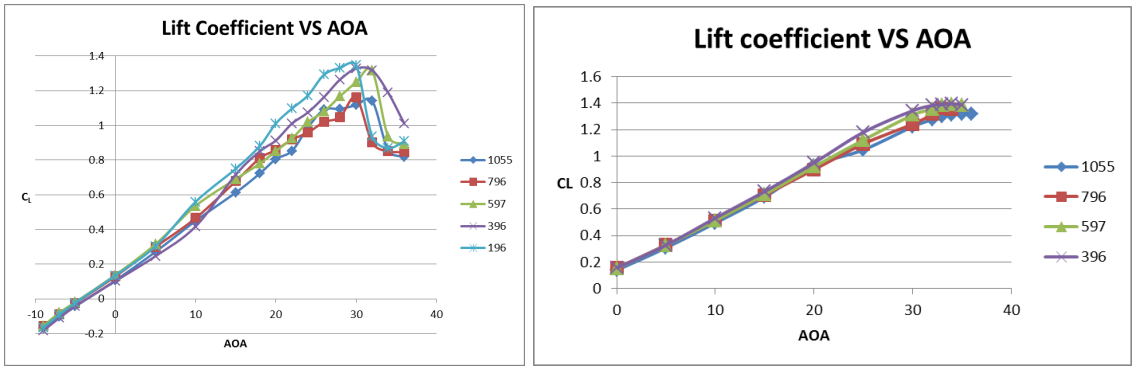
http://www.imavs.org/

attack. At low AOAs, both measurements and CFD predictions indicate that the wingtip thickness has insignificant influence on aerodynamic performances.



(a) Experiment (b) CFD

Figure 7: Comparison of lift and drag coefficients at Re = 150,000 (Blue line: CFD results, Red line: Data from ASU)



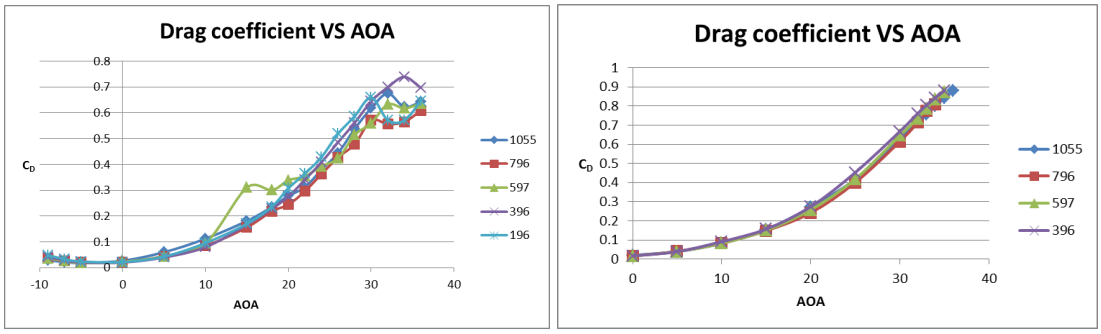
(a) Experiment (b) CFD

Figure 8: Effect of wingtip thickness on lift coefficient

The results for the variations of drag coefficient as a function of angle of attack are presented in Figure 9. It should be noted that the drag forces are quite small and sensitive to the pressure distribution in the leading edge stagnation region, and trailing edge recompression region. The qualitative agreement between the measurements and the predictions is reasonable when this uncertainty in the measurements of (C_D) is factored in. The prediction of transition also influences the prediction of skin friction drag, and can also affect the prediction of drag force. Results show that wings with wingtip of 7.96%c and 5.97%c thickness have less drag force, while other thinner wings (3.96%c and 1.96%c) yield more drag force. Thus, no clear conclusion can be drawn for the effect of wingtip thickness on drag force.

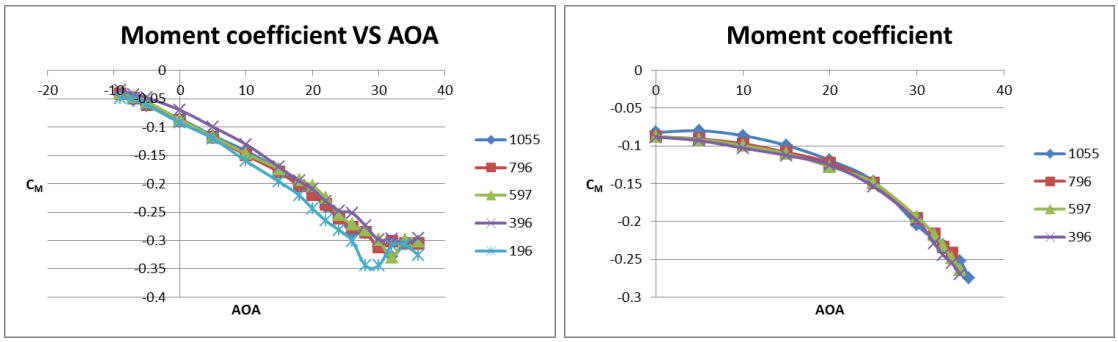
Figure 10 presents the variations of pitching moment coefficient as a function of angle of attack. Results show that wings with thinner tip have more negative pitching moment. However, the longitudinal static stability, $\partial C_M/\partial\alpha$, does not appear to be influenced by the wingtip thickness.

http://www.imavs.org/



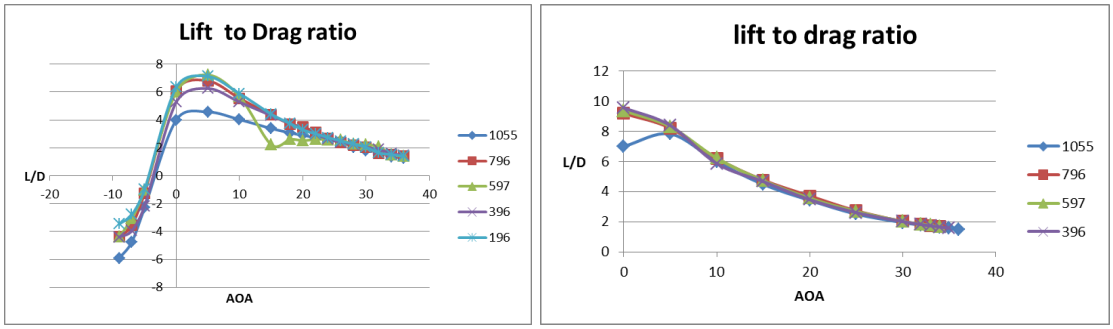
(a) Experiment (b) CFD

Figure 9: Effect of wingtip thickness on drag coefficient



(a) Experiment (b) CFD

Figure 10: Effect of wingtip thickness on pitching moment coefficient



(a) Experiment (b) CFD

Figure 11: Effect of wingtip thickness on L/D

http://www.imavs.org/

Considering the L/D plot shown in Figure 11, both measurement and CFD results clearly reveal a strong influence of wingtip thickness on the L/D at low AOA conditions. Reducing wingtip thickness appears to improve the wing aerodynamic performance only for low AOA cases (up to 20 degrees in experiment, and up to 5 degrees in CFD simulation). For high AOA cases, reducing wingtip thickness is not as effective in improving wing performance as at the low AOA cases. In these cases, they have been found that there are insignificant changes in L/D since lift and drag are noticeably increased in experiments and CFD simulations.

4 Conclusions

The aerodynamics of a series of new design of KUMAV-001 wing has been analyzed using CFD technique and tested in wind tunnel at cruise speed condition. The conditions chosen for detailed study range from low to high angles of attack. Calculations have done using a viscous flow solver, and aerodynamic performance quantities are extracted for comparisons to experiments. Wind tunnel tests have been carried out on a series of wings of varying wingtip thicknesses. Experiment and CFD results show that, at low angle of attack condition, the wing with thin wingtip can produce a net increase in lift-to-drag ratio compared to the baseline wing. At high angle of attack condition, reducing wingtip thickness becomes ineffective in increasing lift-to-drag ratio.

5 Acknowledgement

The authors wish to thank Department of Aerospace Engineering, Faculty of Engineering at Kasetsart University for financial support of this work. The authors would like to thank Mr. Kwanchai Chinwicharnam for preparing *LaTeX* file of this manuscript.

References

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

[2] Jarmo T. Mönttinen. On the performance of micro-aerial-vehicles. Master's thesis, Arizona State University, 2003.

[3] Chinnapat Thipyopas and Nanyaporn Intaratep. Aerodynamics study of fixed-wing mav: Wind tunnel and flight test. In *Proceedings of International Micro Air Vehicle Conference and Competitions 2011 (IMAV 2011)*. Delft University of Technology, Netherlands, September 12-15 2011.

[4] Thomas J. Mueller. Aerodynamic measurements at low reynolds numbers for fixed wing micro-air vehicles. Technical report, VKI, Belgium, September 13-17 1999.

[5] Dragos Viieru, Roberto Albertani, Wei Shyy, and Peter G. Ifju. Effect of tip vortex on wing aerodynamics of micro air vehicles. *Journal of Aircraft*, 42(6):1530–1536, November 2005.

[6] Rémi Blanc, Jean-Baptiste Vallart, Boris Bataillé, and Jean-Marc Moschetta. Thick airfoils for low reynolds applications: improvement of a biplane micro aerial vehicle. In *European Micro Air Vehicle Conference and Flight Competition (EMAV 2008)*, Braunschweig, Germany, July 2008.

[7] Jewel B. Barlow, William H. Rae, and Alan Pope. *Low-Speed Wind Tunnel Testing*. John Wiley & Sons, 1999.

http://www.imavs.org/

Aerodynamic Characteristics of a Low Aspect Ratio Wing and Propeller Interaction for a Tilt-Body MAV

Kwanchai Chinwicharnam[1]*, David Gomez Ariza[2]†, Jean-Marc Moschetta[2]‡ and Chinnapat Thipyopas[1]§

[1] Kasetsart University, Bangkok, Thailand
Chinwicharnam.Kwarchai@isae.fr
[2] Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse, France
Jean-Marc.Moschetta@isae.fr

http://www.imavs.org/

Abstract

An experimental investigation of the interaction of a propeller-wing configuration for a tilt body MAV VTOL was performed in the low speed wind tunnel. This study's primary objective is to present the effect of the interaction between a low aspect ratio wing and propeller for the range of incidence in transition between horizontal and vertical flight. During the transition from horizontal flight to vertical flight or vice versa, the flow patterns seen by the wing are the result of the combination between the free-stream and the propeller flow. This was reflected in the change of the aerodynamic forces and moments of the wing. The model is a tractor configuration which is a Graupner propeller and wing with aspect ratio 1, an airfoil NACA 0012. All tests were conducted at low speeds in a range from 2 to 8 m/s. In order to simulate the transition flight of a MAV VTOL a range of incidence from -10 to 90 degree was used. The results show that the flow patterns of the propeller certainly improve the aerodynamic characteristics of the wing, increase lift and delay stall angle with respect to the flight path of the MAV.

1 Introduction

The interaction between wing and propeller was considered for improving the aerodynamic performance of MAV. The propeller propulsive system, which is the main power for flight operation, is a problem of MAV aerodynamic because the motor and propeller are located across the free stream; and correspondingly the propeller generates slipstream. The position of propeller influence on the wing boundary layer characteristics such as: laminar flow extension and transition, laminar separation bubbles, and reattachment and turbulent separation. This has been found and compared the different between tractor and pusher configuration by Catalano[1] which states that pusher propeller inflow affects the wing characteristics by changing the lift, drag, and also the boundary layer transition and separation point. The propeller slipstream is a cause of variable flow property around surface of MAV and induces the free stream velocity for MAV. To understand the behavior of the flow seen by the wing, the experiment was organized in order to analyze the difference between the isolated wing and the mounted propeller-wing. The effect of propeller wash or slipstream on the wing aerodynamic during transition flight was the focus and the fixed-wing tractor configuration was considered.

*Master student of Kasetsart University
†PhD. student of ISAE (davinciig@yahoo.com)
‡Professor, Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse, France.
§Advisor, Department of Aerospace Engineering, Kasetsart University, Bangkok, Thailand. (fengcpt@ku.ac.th)

http://www.imavs.org/

Similar researches has shown that the slipstream effect to the stall delay, lift augmentation, drag increase, and reduced aerodynamic efficiency [2][3]. This research studied and supported the Mini-Vertigo MAV configuration was a co-axial propeller. Maxime [4] studied the characteristic longitudinal flight behavior during an equilibrium transition between vertical/horizontal flight modes and were investigated for enhancement of the longitudinal control of the MAVion (a tractor configuration MAV of ISAE). This test found the interaction wing and propeller and the model has 2 propellers. Deng et al.[5] studied the propeller-wing interaction both the experiment and numerical method. It was found that the slipstream has a significant influence on the pressure distribution on the wing surface, as well as, explained the pattern of wing-tip vortex in different of the angle of attack with rotary propeller. However, the test is limited at low angle of attack. Therefore, the aerodynamic part of interaction wing and propeller is essential to study and understand the performance of MAV during trasition flight, especially. Moreover, the wing wash effect which several researches neglected has been considered in this paper. This current research investigates the effect of propeller-wing interaction in a tractor configuration by performing an experiment. A three dimensional NACA 0012 rectangular wing and Grauper Super Nylon propeller was performed in the Supaero low speed wing tunnel (SabRe). The isolated wing and propeller were conducted and then the mounted propeller-wing was considered at the same condition.

2 Test Subject

The model in this study is a tractor configuration. The detail of model is in Figure 1. The test was achieved at ISAE by the SabRe wind-tunnel. The model was investigated both at a low incident angle and high incident angle; thus the range of angle of attack varies from -10 to 90 degrees. The model was also designed with a plain flap in order to study the flap deflection effect. In this experiment was focused at the low Reynolds Number(Re) . Moreover, the installations propeller position was also studied in this research.

Information	Value
Airfoil	NACA 0012
Wingspan	0.3 m
Wingchord	0.3 m
Planform shape	Rectangular
Wing area	0.09 m^2
AR	1
Flap type	Plain flap
Flap area	0.03 m
Wing material	SLS Materials
Motor	Brushless PJS 3D 550
Propeller	Graupner 8x6"
Motor support	Aluminum

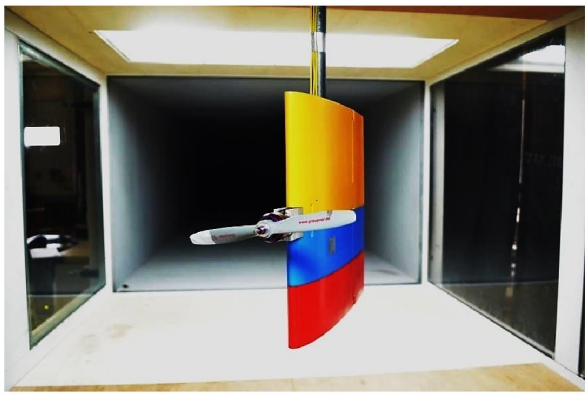


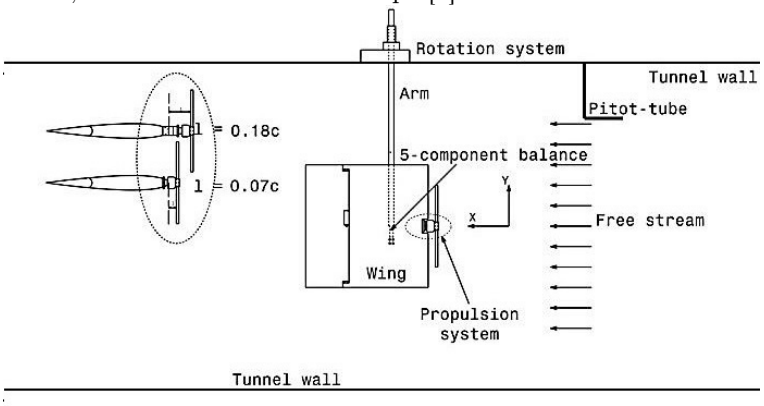
Figure 1: Model information and tractor configuration model in SabRe

3 Experimental Setup

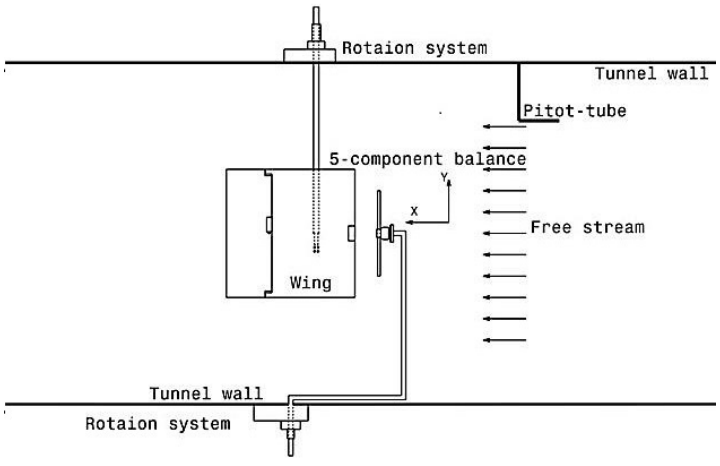
The experiment was performed using four different models: a singular propeller model was achieved by David[6], a wing alone model, MPROWM (Mounted propeller on wing model) and SPROWM (Separated propeller on wing model). The model was installed in the closed-circuit low speed wind tunnel of ISAE (SabRe) with test section 1.2m x 0.8m and 2.4m long.

The model was positioned vertically. The propeller was moved by the aluminum motor support wedges. The propeller (l) was set in position at 7%c and 18%c ahead of wing leading edge(LE). The model was turned from -10 to 90 degree by the using an automatic rotated motor, in order to measure the angle of attack effect of the models. The flap deflections, in a range from -10 to +20 degrees, were moved by a digital servo mechanism inside the wing. The propeller speeds were fixed at 5,000 RPM. The free stream velocity was set in a range from 2 to 8 m/s.

The isolated wing, MPROWM and SPROWM were carried out in the SabRe wind tunnel shown in Figure 2(a) and 2(b). The forces and moments were measured by the internal five-component Micro Sting Balance aerodynamic which limited at 10 N forces and 0.5 N-m moments. It was made of high strength 35NCD16 alloy steel which calibrated by Thipyopas [7]. All connectors were connected to the National Instrument; PXI-6281 and PXI-6229 card were used to measure the voltage-usage for balance and current-usage for rotational propeller, respectively. Data was collected through the Labview program at 1,000 Hz and recorded every 10 second. Moveover, there are 10 samples for each AOA were used in order to get the accurate data. Note: all of data has been corrected through the wind tunnel wall effects, which is the method of Pope [8].



(a)



(b)

Figure 2: a) Wind-tunnel tests set up of MPROWM, b) Wind-tunnel tests set up SPROWM

http://www.imavs.org/

4 Results

The aerodynamic coefficient in this paper can be calculated as:

$$C_L = \frac{L}{\frac{1}{2}\rho V^2 S}, C_D = \frac{D}{\frac{1}{2}\rho V^2 S}, C_M = \frac{M}{\frac{1}{2}\rho V^2 Sc}, C_X = \frac{X}{\frac{1}{2}\rho V^2 S}$$

4.1 Mounted Propeller on Wing Model (MPROWM)

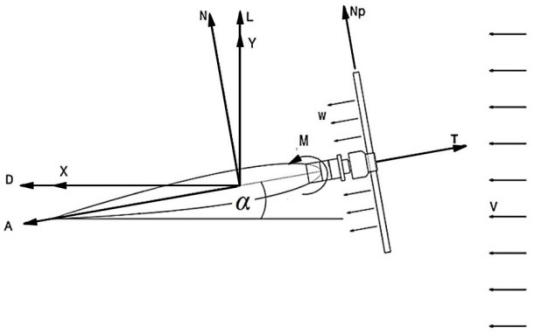


Figure 3: Free body diagram of MPROWM

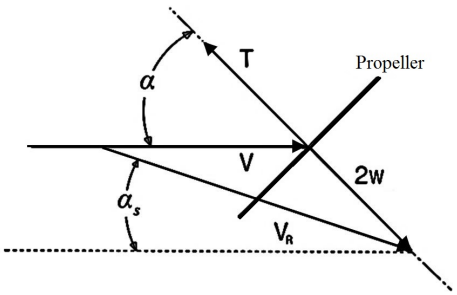


Figure 4: Velocity triangle of fully accelerated slipstream

The balance was set inside the wing and then it was calibrated forces and moments by the standard weight. The AOA and flap deflection angle was calibrated as well. The error is very small and can be ignored. The propeller was mounted with the wing at 7%c ahead of the wing LE. The test fixed the rotating speed at 5000 RPM. The force vector is considered in terms of free body diagram in Figure 3. In the combination of wing-propeller as the MPROWM; the resulting aerodynamic forces are generated by the combination of the singular propeller forces, the single wing forces, and the effect of the propeller wash over the wing. The main effect of the propeller wash over the wing can be seen by the changing effect of AOA on the wing. This AOA can be calculated by using the method in suggested by McCormick[9] that the propeller can make the induced velocity ahead of the wing as shown the velocity triangle of fully accelerated slipstream in Figure 4. In Figure 5(a) shows the ratio of the induced velocity for the general case to the induced velocity in hover case increases with the increment of AOA. And a cause of the effective wing angle of attack (AOA_w) is the slipstream resulting velocity (V_R) which is the combination between the forward velocity and induced velocity. The evolution of AOA_w as a function of the MPROWM AOA is plotted for V= 8 m/s in Figure 5(b) and the estimation of wing AOA is simply equal to:

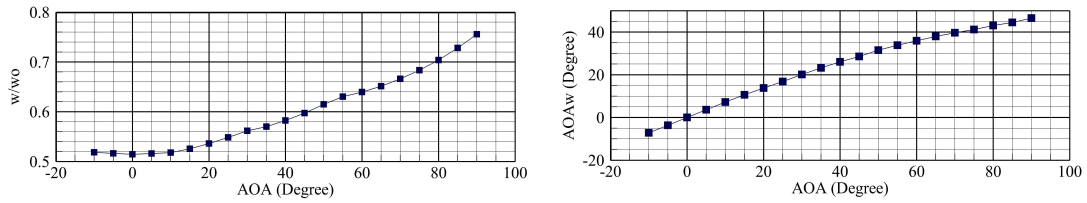
$$AOA_w = AOA - AOA_{slipstream} \quad (1)$$

4.2 Wing and Propeller Wash Effect

4.2.1 Lift Coefficient

In Figure 6(a), the variation of wing prop-on (MPROWM) and wing alone lift with AOA are compared. The figures show the propeller slipstream has a strong stall-delay effect and they have the perfect linear

http://www.imavs.org/

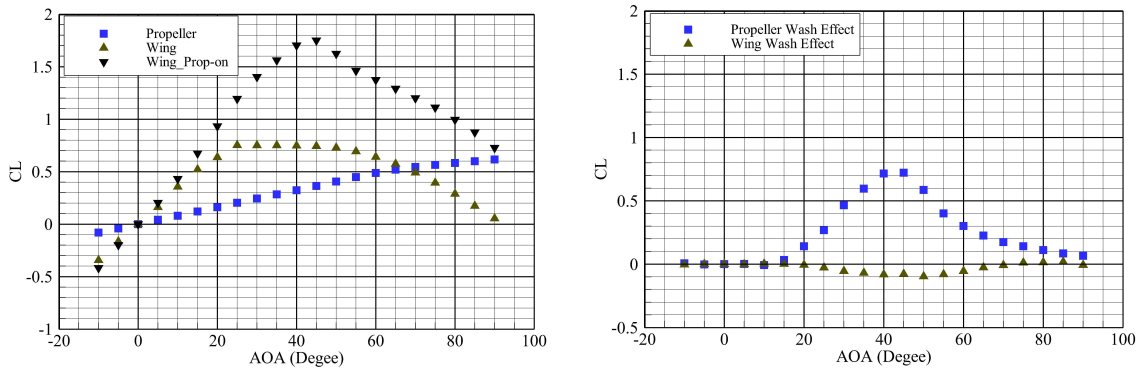


(a) (b)

Figure 5: Propeller slipstream effect when V=8m/s RPM5000: (a) induced velocity, (b) wing angle of attack

slope at post-stall. The stall angle moves from 25 degrees to 40 degrees, as well as, the maximum wing prop-on lift increases by 130%. Consider the generated lift coefficient in each part of MPROWM, which can be derived as:

$$C_{Lwing-prop} = C_{Lwing} + C_{Lprop} + \triangle C_{Lwing/prop} + \triangle C_{Lprop/wing} \tag{2}$$



(a) (b)

Figure 6: (a) Propeller, Wing and MPROWM lift coefficient versus angle of attack, (b) Propeller/Wing wash effect lift coefficient versus angle of attack : when V = 8 m/s RPM 5000

Consider the stall angle of the wing prop-on is at 40 degrees, but the true wing AOA is 25 degrees from Figure 6(b) that is also the stall angle of the wing prop-off. This proves Equation 1 works. Notice the including lift coefficient of wing prop-off and on is less than MPROWM lift coefficient. This shows that something promotes the wing and propeller interaction effect. Hence the SPROWM was performed and investigated the propeller wash effect and wing wash effect shown in Figure 6(b). It seems the MPROWM gains lift from wing and propeller only at -10 to 15 degrees. Subsequently, the propeller slipstream has a strong effect on the model up to its maximum of 40-45 degrees, which is the stall angle. These results confirm again that the propeller slipstream can develop the wing boundary layer characteristics such as: laminar separation bubbles, reattachment and turbulent separation; as the suggestion by Catalano [1]. A cause of the stall delay is supposed that the flow reattaches to the wing surface again due to the propeller slipstream (Propeller-wash effect) $\triangle C_{Lwing/prop}$. Therefore the propeller wash effect has been connected to the cause of the MPROWM lift generated. However the wing wash effect $\triangle C_{Lprop/wing}$ which is resistant to the MPROWM lift has also been observed.

http://www.imavs.org/

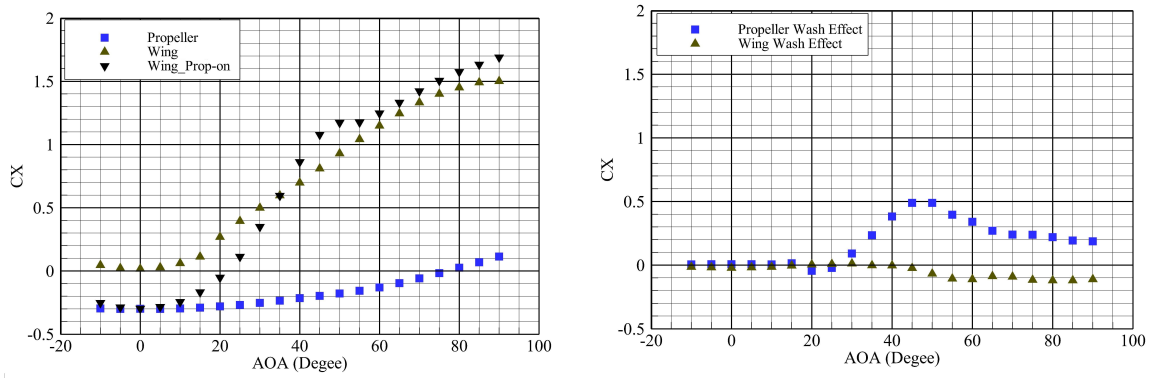
It is small effect to propeller, the maximum is about 14% of Max. $\triangle C_{Lwing/prop}$ at stall angle, and negative value during stall of MPROWM as well.

4.2.2 Total Longitudinal Force Coefficient

In Figure 7(a) shows C_X wing alone increases continuously with AOA, until the propeller is on. The maximum C_X is still at 90 degrees and rise 13% as a result of the slipstream. In the beginning (-10 to 15 degrees), the offset of wing prop-on and wing alone C_X are completely influenced by the propeller. After this the value of MPROWM C_X is larger than the propeller C_X minus wing prop-off C_X. Again the negative value of C_X is with respect to the forward flight direction. Therefore, consider the results generated by C_X in each part of MPROWM.

$$C_{Xwing-prop} = C_{Xwing} + C_{Xprop} + \triangle C_{Xwing/prop} + \triangle C_{Xprop/wing} \tag{3}$$

In Figure 7(b) shows that neither the propeller wash nor wing wash effect at -10 to 25 degrees. Such is the reason of MPROWM C_X value is equal wing prop-off C_X plus propeller C_X. Then, the propeller-wash effect C_X increases, which identifies the slipstream as making the incremental increase of MPROWM C_X. Hence, the propeller slipstream (Propeller wash effect) prompts the drag for the wing which indicates a concave-up parabolic measurement in the pre-stall and concave-down after the stall. However, after stall angle is found, the result is the wing-wash effect C_X. It promotes the reducing of MPROWM C_X.



(a) (b)

Figure 7: (a) Propeller, Wing and MPROWM total longitudinal force coefficient versus angle of attack, (b) Propeller/Wing wash effect longitudinal force coefficient versus angle of attack : when V = 8 m/s RPM 5000

4.2.3 Pitching Moment Coefficient (LE)

Pitching moment is considered at LE, thus the negative is pitching down and positive is pitching up. In Figure 8(a), the maximum $C_{m(LE)}$ rises 150% due to propeller. The propeller slipstream has still effect to the wing pitching moment. In Figure 8(b) shows that the slipstream has been affected since 15 degrees. It seems the wing can generate more lift by propeller slipstream, thus the MPROWM has more pitching down automatically. The curve of propeller slipstream (Propeller wash effect) is concave-down parabolic measurement in the pre-stall and concave-up after the stall. For the wing wash effect decreases gradually after 15 degrees. It seems the propeller wash and wing wash have an effect to MPROWM $C_{m(LE)}$. Again the MPROWM $C_{m(LE)}$ can be separated for each part as:

http://www.imavs.org/

$$C_{m(wing-prop)} = C_{m(wing)} + C_{m(prop)} + \triangle C_{m(wing/prop)} + \triangle C_{m(prop/wing)} \quad (4)$$

where $C_{m(prop)} = C_T \cdot r + C_{Np} \cdot l$. The average C_T is close to the propeller center (r), but to r is very small. Thus $C_T \cdot r$ is considered to be zero. l is the propeller position from the wing LE.

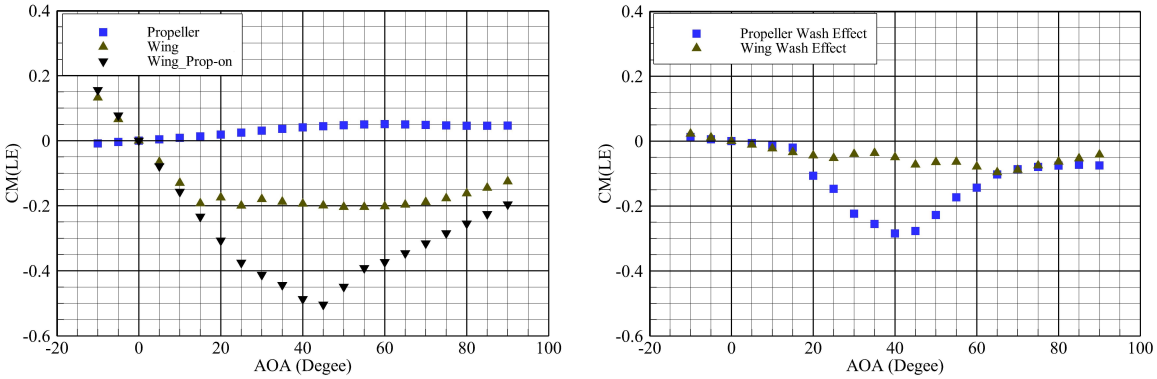


Figure 8: (a) Propeller, Wing and MPROWM pitching moment coefficient versus angle of attack, (b) Propeller/Wing wash effect pitching moment coefficient versus angle of attack : when V = 8 m/s RPM 5000

4.3 Streamwise Propeller Position Effect

The position of propeller was changed from 7% to 18% of mean aerodynamic chord ahead of wing LE in order to investigate what effect of the propeller slipstream has on the wing. Figure 9, found that the slope over the stall of the wing for a propeller located at 18%c decreases immediately but the propeller located at 7%c gradually reduces . This seems that the propeller, which is closer to the wing, can control the flow around the wing and improve it. That propeller position may also help the flow attach more to the wing surface. The flow after stall angle improves by propeller-wash; the blue line, which is the influence of propeller wash on the wing, can be explained clearly by the C_L,C_X curve are smooth concave-down parabolic.

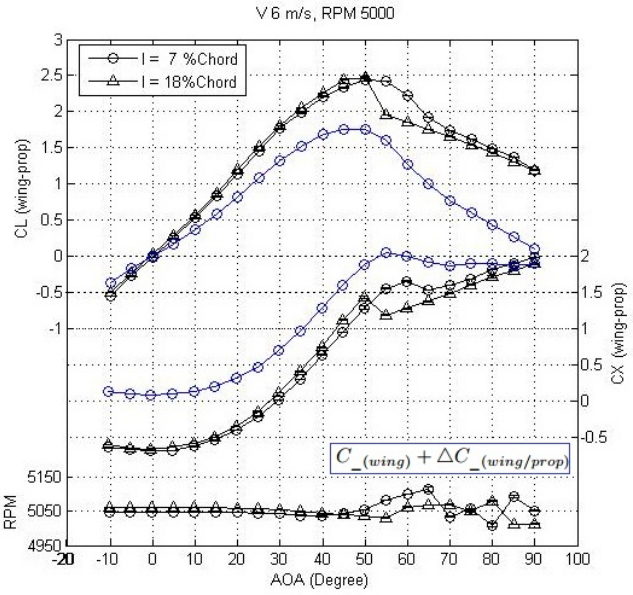


Figure 9: Lift, Drag coefficient and RPM vs AOA

The best performance of the wing is obtained when the propeller is installed close to the wing as the previous studies as the one presented in [10]showed the results at low AOA. Moreover, the current results show that for a high AOA and the results are same case. It should be noted that there are small difference of RPM in each AOA, but there are small energy results which can be ignored.

http://www.imavs.org/

http://www.imavs.org/

4.4 Flap Deflections Effect

In order to investigate the flap deflections effect, the MPROWM and SPROWM were designed with a flap at -10 to 20 degrees. In above part can see the effect of propeller slipstream takes place on the wing. In Figure 10 shows the influence of propeller slipstream on the wing in terms of variation of flap deflections. The aerodynamic values are improved due to the propeller slipstream develops the boundary layer on the wing surface and keeps laminar flow in higher incident angle. The wing C_L and C_X increase when the flap deflection is positive. Conversely, the wing C_L-C_X decrease when the flap deflection is negative. In conclusion, the changing of wing C_L and C_X in each ± 10 degrees of flap deflections are ± 0.35 and ± 0.3 respectively. The flap deflections have no an influence on the wing pitching moment at the leading edge($C_{m,LE}$). We found that the wing lift increases when the flap deploys in positive, and the the pitching moment at 0.3c as well. Therefore, it is normally that the wing pitching moment at LE in variation of flap deflections is not different at post-stall. During the first and second stall angle (20-35 degrees), the curves swing because of the severation flow and the strong turbulent flow on the wing surface. Moveover, the flap of MPROWM is out of control the wing aerodynamic after stall angle because of the slipstream effect.

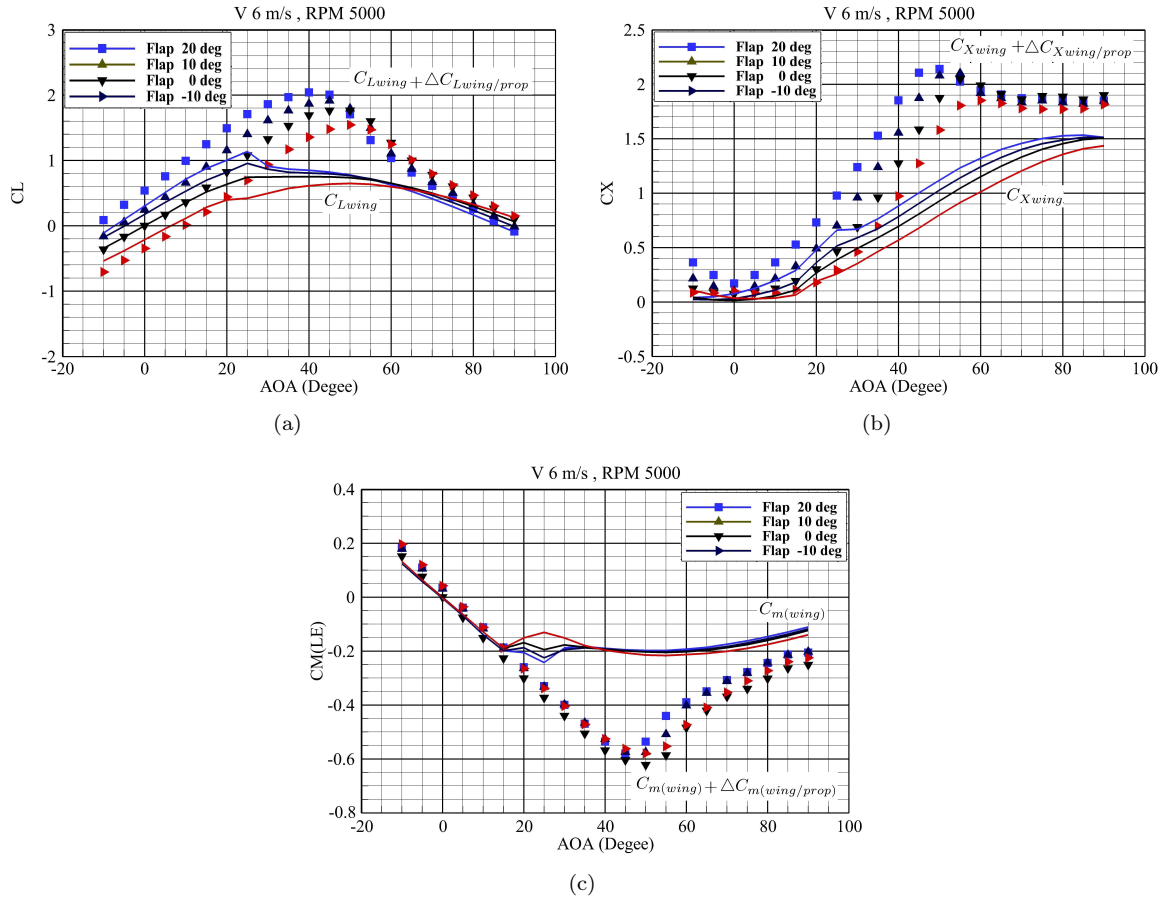


Figure 10: Flap deflection effect on the MPROWM vs AOA: (a) lift coefficient, (b) total longitudinal force coefficient, (c) pitching moment coefficient at leading edge

5 Conclusion and Future Work

Due to multi-function of MAVs, it can tilt body from horizontal to vertical flight, as well as hover. Thus it is important to know to aerodynamic characteristics which influence flight control while flying. In order to investigate the aerodynamic characteristics, we focused on the interaction between a wing and propeller. Additionally, this research is different from the other research done with the models. The models of this research are applied by the basic configuration such as: the wing in NACA 0012, the electronic propeller which is general to explain the aerodynamic performance of interaction between wing and propeller for tilt body MAVs. The models were tested in SabRe wind tunnel and found that

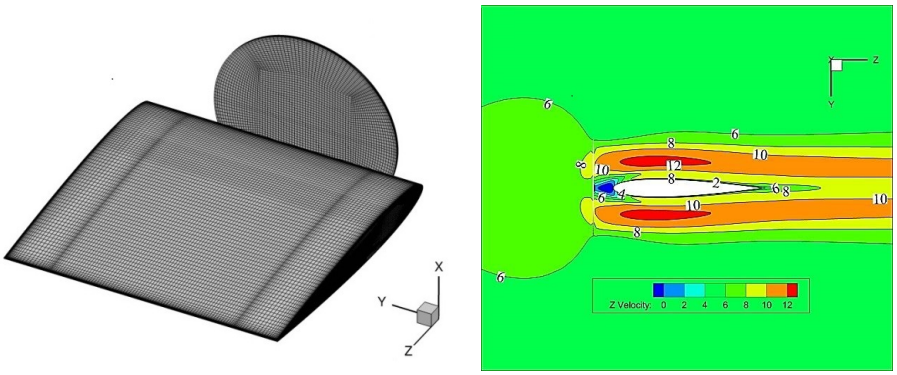
- The MPROWM aerodynamic forces and moment are not only generated by the singular wing and propeller, but also the slipstream effect.
- Moreover, the slipstream is divided to propeller-wash effect and wing-wash effect. The propeller-wash has the large effect, but the wing-wash has some small effect to the wing. The influence of propeller-wash on the wing develops the boundary layer and keeps laminar flow in higher incident angle. These effects increase the wing performance and delay stall angle
- The installations with close to the wing are more efficient than those farther. Especially after the wing stall, the wing boundary layer has more the reattach flow, and as well the wing lift has a smooth downward motion.
- Maximum MPROWM lift and drag increases with positive flap deflection and decreases with negative flap deflection at post stall. And slipstream propeller improves the flow around the wing

The experiment data of this study can only explain the interaction between wing and propeller of tilt body MAVs in terms of aerodynamic coefficient. Therefore the future work will use Computational Fluid Dynamic (CFD) methodology. In order to achieve the level of detail to describe the phenomena flow that occurs in the propeller-wing interaction, as well as, confirm the assumption of experimental study about the flow, the numerical methodology uses the FLUENT. This intends to explain the propeller slipstream which is the main point of flow around the wing. The k-ε RNG turbulent model is applied. The main interest is that the RNG model in FLUENT provides an option to account for the effects of swirl or rotation. The Pressure-Velocity coupling solves to get the convergence by the SIMPLE C algorithm. The propeller is assumed to an actuator disk. Moreover UDF is called for the actuator disk boundary condition which the velocity polynomial profile function of propeller is suggested by Rosen [11]. A structured C-grid type is used and shown in Figure 11(a). The first case at zero AOA has been done and shows the axial velocity contour plot in Figure 11(b), which also found the axial velocity increases at propeller downstream.

6 Acknowledgments

The author is indebted and grateful to Chinapat Thipyopas of the Aerospace Engineering of Kasetsart University, Thailand who introduced the research, and Jean-Marc Moschetta of Institut Supèrieur de l'Aèronautiaue et de l'Espace, France who recommended a great deal of this research as well as providing the place and tools for the test. David Gomez Ariza, guided the processing of research and helped solve problems during the test, as well as giving suggestions and exchanges ideas. The author appreciates Rèmi Chanton, Sebastien Prothin and other members of SUPAERO for experimental setup preparation and their involvement. Finally, the author would like to thank Ryan Randall, his previous works helped give invaluable information to this writer.

http://www.imavs.org/



(a) (b)

Figure 11: (a) Structure C-grid of wing and actuator disk, (b) Axial velocity contour plot when the MPROWM AOA 0 degree and freestream velocity 6 m/s

References

[1] F.M. Catalano. On the effects of an installed propeller slipstream on wing aerodynamic characteristics. *Acta Polytechnica*, 44(3):8–14, 2004.

[2] R. Randall, C.-A. Hoffmann, and S. Shkarayev. Longitudinal aerodynamics of a vertical take off and landing micro air vehicle. *Journal of Aircraft*, 48(1):166–176, January-February 2011.

[3] R. Randall, S. Shkarayev, G. Abate, and B. Judson. Longitudinal aerodynamics of rapidly pitching fixed-wing micro air vehicle. *Journal of Aircraft*, 49(2):453–467, March-April 2012.

[4] M. Itasse, J.-M. Moschetta, Y. Ameho, and R. Carr. Equilibrium transition study for a hybrid mav. *International Journal of Macro Air Vehicles*, 3(4):229–246, November 2011.

[5] S. Deng, B.W. van Oudheusden, T. Xiao, and H. Bijl. A computational study on the aerodynamic influence of a propeller on an mav by unstructured overset grid technique and low mach number reconditioning. *The Open Aerospace Engineering Journal*, 5:11–21, 2012.

[6] D.G. Ariza. *Study of the Sensitivity to the Lateral Wind of a Mini Unmanned Aerial Vehicle with VTOL flight capabilities*. PhD thesis, ISAE, France, 2013.

[7] C. Thipyopas and J.-M. Moschetta. Experimental analysis of a fixed-wing vtol mav in ground effect. *International Journal of Macro Air Vehicles*, 2(1):33–54, March 2010.

[8] A. Pope, J.B. Barlow, and W.H. Rae. *Low-Speed Wind Tunnel Testing*. John Wiley & Sons Inc, New York, 2011.

[9] B. W. McCormick. *Aerodynamics of V/STOL Flight*. Dover, New York, 1967.

[10] V. Leo. *Propeller Wing Aerodynamic Interference*. Delft University of Technology Netherlands, Netherlands, 2005.

[11] A. Rosen and O. Gur. Novel approach to axisymmetric actuator disk modeling. *Journal of AIAA*, 46(11):2914–2925, November 2008.

[12] B. Bataille, J.-M. Moschetta, D. Poinsot, and C. Berard. Development of a vtol mini uav multi-tasking missions. *The aeronautical Journal*, 113(1140):87–98.

http://www.imavs.org/

Vector thrust multi-rotor copter and its application for building inspection

C.-H. Kuo[1], C.-M. Kuo[1], A. Leber[1] and C. Boller[1,2]

[1] Saarland University, Mat. Sc. & Technol. Dept., Chair in NDT & QA, Saarbrucken, Germany
[2] Fraunhofer Inst. for NDT (IZFP), Saarbrucken, Germany

Abstract

Multi-rotor copters are becoming increasingly popular for research and experimen-tation of air vehicle control, aerial imaging (photo/video) and even the hobby market, due to its excellent performance and mechanical downscaling as well as cost when compared to its conventional full scaled and possibly even manned siblings. The re-search groups at both Fraunhofer IZFP and Saarland University have carried out ex-periments, using a multi-rotor copter with an off the shelf camera for building in-spection (aerial photo) for years [1]. With possibly all of the conventionally built ro-tary wing micro aerial vehicles (MAV) the translation of the vehicle in any direction will cause the vehicle to tilt its body, which leads to distortion of images/photos even with an integrated gimbal system as a tool for image stabilization. In order to keep the vehicle in a stable, horizontal position even during translation, a novel vector thrust system has been designed, analyzed through use of numerical simulations, and proven by means of hardware realization and experimental flights. The first part of this paper discusses the modeling of the vector thrust system using a plus- and x-configuration. The second half of the paper consists of the numerical simulation to prove the concept. The last part describes demonstration of the concept with real hardware and experimental flight results.

Glossary:
F_x, F_y, F_z**=Force in X,Y,Z direction (body axis)**
w_i**=rotational speed of motor i**
ϕ**,** θ**,** ψ**=Roll,Pitch and Yaw angle (attitude)**
cϕ**,** θ**,** ψ**.s**ϕ**,** θ**,** ψ**= cos**ϕ **or** θ **or** ψ**.sin**ϕ **or** θ **or** ψ
b=Lift constant
d=Drag constant

http://www.imavs.org/

1 Introduction

An enhanced amount of civil infrastructure buildings (including roads and bridges) has become an issue with regard to the ageing process and hence a resulting life cycle management. According to the Federal Highway Administration (FHWA), nearly 70% of all bridges and roads need to be inspected regularly in the US [2, 3, 4, 5, 6]. This in-spection is based on purely visual methods and is completed by inspection personnel present on site [7, 8, 9]. However, apart from bridges, no institution is estimating the number of required inspections and methods with regard to infrastructures such as government buildings or factories. However, the inspection of certain structures can be dangerous for inspectors to be on site, as some locations on the structure may only be accessed under extreme circumstances, such as very slender, high towers [10, 11, 12, 13]. Since most inspections refer to visual inspection (95%) a robot, equipped with a digital camera, instead of a human inspector is proposed

to be used. An octocopter is a fairly popular solution and was hence used throughout this project, as it offers sufficient payload capacity, is small in size, and most importantly, is comparatively simple with regard to the mechanical structure. (Fig. 1)



Diameter	1.02 m
Take-Off Weight	2.5 kg
Endurance	< 20 min
Max. Payload	2 kg

Figure 1: Octocopter used and technical data

The first inspection task took place at the Fraunhofer IZFP building in Saarbrucken/Germany (Fig. 2). The total inspection took less than 8hr (which con-sisted of time for the actual flight and photo checking). The results of the inspection, in form of images of each facade combined in a 3D model, can be seen in Fig. 3



Figure 2: **Left**: Aerial view of Fraunhofer IZFP with rendered line indicate inspection facade. **Right**: Front view of the Fraunhofer IZFP.

The inspection mission was significantly faster than a conventional inspection would have been. There were around 4GB of photo material collected for each facade of the building. However, only around 10% of the photos were selected for the image processing in the end. The rest was considered as 'noise' since different of the imag-es showed the same however with a slight variation in attitude (Fig. 4).
Even though the octocopter was equipped with a gimbal system, attitude fluctua-tion still affects the image captureD, therefore reducing the 'noise' resulting from attitude variation becomes important to further enhance the imaging process.

2 Vector thrust propulsion for robot (reduction of attitude noise while in action)

Any translation in x, y and z direction of the MAV will require attitude change [14, 15, 16], due to the fact that the motions are all controlled by a difference in rotation speed of the motors (Fig. 5).

http://www.imavs.org/

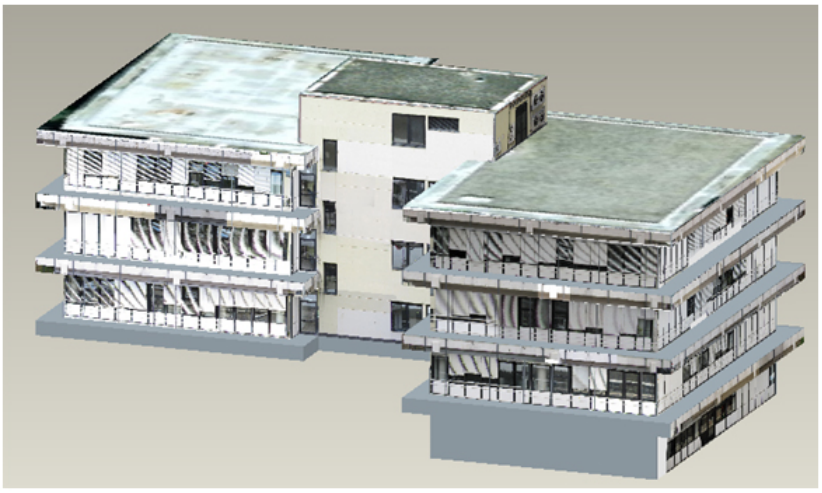


Figure 3: 3D CAD model of inspected building (Fraunhofer IZFP)

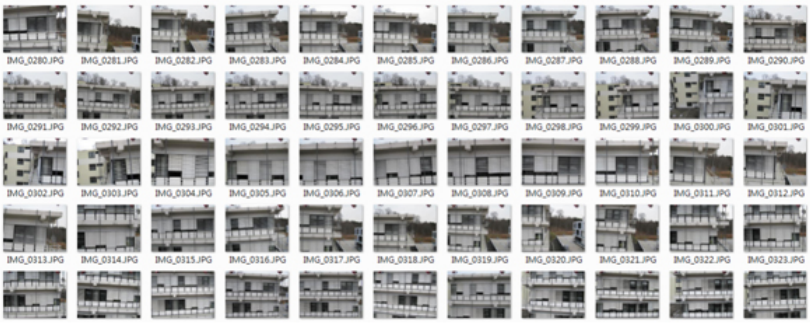


Figure 4: Few % of photos taken as an example to show variation in attitude

This attitude change is a source of noise during the image capturing process (even with a camera gimbal stabilization, attitude change still has a very large effect). However, translation in any x, y and z direction is required for the inspection. Therefore a vector thrust system is proposed in order to remove atti-tude changes when a motion is applied to the multi-rotor vehicle.

Vector thrust in the type of rotary wing vehicle shown above can be realized by twisting the rod along which the rotor is attached, hence around the x- and y-axis respectively when taking the definitions provided in Fig. 5(left). To describe the di-rection of motion of the aerial vehicle itself the position of rotors versus the direction of motion of the aerial vehicle itself becomes important. In the case of four rotors and when the aerial vehicle moves in the direction of either the x- or y-axis within the x-y-plane the configuration will be called a plus-configuration while when the aerial vehicle moves in a direction between the x- and y-axis it will be called an x-configuration. Comparing the two configurations it already becomes obvious that efficiency with respect to vehicle stabilization becomes less in the plus-configuration since two of the vector thrust rotors along the direction of motion basically do not show any effect. Within the following the analytical modeling of the two

http://www.imavs.org/

configura-tions will be described before applying the models along simulations.

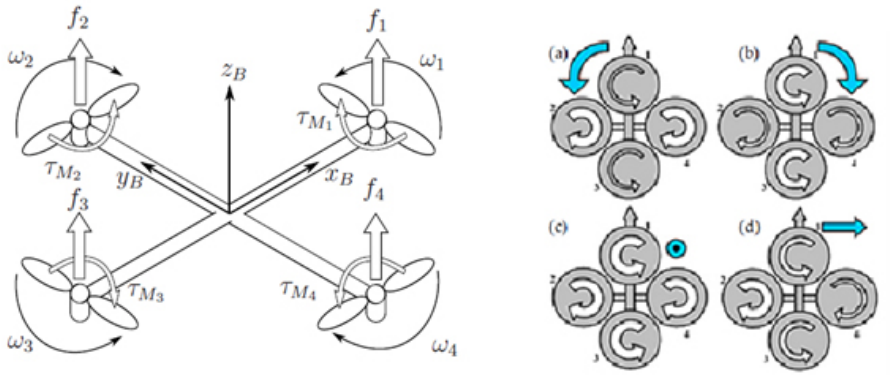


Figure 5: **Left**: Free force diagram of the multi-rotor robot (quad). **Right**: Motion diagram, a) yaw anti-clockwise, b) yaw clockwise,c) increase altitude (movement in Z direction only),d) roll positively to create motion in y direction (negatively).

3 Modeling of plus frame vector thrust

Lift force on each motor is

$$L_i = F_i b w_i \tag{1}$$

Torque of each motor is

$$\tau_i = d w_i \tag{2}$$

6 DOF generalized vector q

$$q = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T \tag{3}$$

The Lagrangian is obtained by modeling the energy of the system which is the differ-ence between kinetic and potential energy.

$$L = T - U \tag{4}$$

Where T is the kinetic energy and U is the potential energy. The final expression is:

$$L(q,q) = \frac{\dot{1}}{2} m(\dot{x} + \dot{y} + \dot{z}) + \frac{1}{2}(I_{xx}\dot{\phi}^2 + I_{yy}\dot{\theta}^2 + I_{zz}\dot{\psi}^2) - mgz \tag{5}$$

If an external generalized force is applied:

$$F = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} \tag{6}$$

$$F = [F_\xi \ \tau]^T \tag{7}$$

http://www.imavs.org/

$$F_\xi = [F_x \ F_y \ F_z]^T \tag{8}$$
$$\tau = [\tau_\phi \ \tau_\theta \ \tau_\psi]^T \tag{9}$$

Where :

$$F_x = \sum F_{m2\&4} = -b(w_2^2 \ + \ w_4^2)sin\theta_s$$
$$F_y = \sum F_{m1\&3} = -b(w_1^2 \ + \ w_3^2)sin\phi_s$$
$$F_z = \sum F_{m2 \ to \ 4} = -b[(w_2^2 \ + \ w_4^2)cos\,|\theta_s| + (w_1^2 \ + \ w_3^2)cos\,|\phi_s|] \tag{10}$$

The PDE of $\frac{d}{dt}\frac{\partial L}{\partial \dot{q}}$ and $\frac{\partial L}{\partial q}$ can be expressed as follows:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \\ I_{xx}\ddot{\phi} + \dot{I}_{xx}\dot{\phi} \\ I_{yy}\ddot{\theta} + \dot{I}_{yy}\dot{\theta} \\ I_{zz}\ddot{\psi} + \dot{I}_{zz}\dot{\psi} \end{bmatrix}$$

and

$$\frac{\partial L}{\partial q} = \begin{bmatrix} 0 & 0 & -mg & 0 & 0 & 0 \end{bmatrix}^T \tag{11}$$

Now, the force from the body frame must be translated to a global frame by using the rotational matrix R.

$$\frac{\partial L}{\partial q} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & \text{-}s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\psi \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi + s\phi c\psi & c\phi c\psi \end{bmatrix} \tag{12}$$

Note:

$$F_\xi = R.F_{rotor} \tag{13}$$

$$\tau_\phi = (F_{2z} - F_{4z})l + (\tau_2 + \tau_4)sin\theta_s \tau_\theta = (F_{3Z} - F_{1Z})l + (\tau_1 + \tau_3)sin\theta_s$$
$$\tau_\psi = (\tau_1 + \tau_3)cos\,|\phi_s| + (\tau_2 + \tau_4)cos\,|\theta_s| \tag{14}$$

Where $(\tau_2 + \tau_4)sin\theta_s$ & $(\tau_1 + \tau_3)sin\phi_s$ are very small turns, which can be assumed to be zero. Thus, the Euler-Lagrange equation with (7, 8, 9, 10, 11, 13 and 14) together yield to:

$$\ddot{q} = \begin{pmatrix} \frac{F_x c\theta c\psi + F_y c\theta s\psi - F_z s\theta}{m} \\ \frac{F_x(s\phi s\theta c\psi - c\phi s\psi) + F_y(s\phi s\theta s\psi - c\phi c\psi) - F_z s\phi c\theta}{m} \\ \frac{F_x(c\phi s\theta c\psi - s\phi s\psi) + F_y(c\phi s\theta s\psi - s\phi c\psi) - F_z c\phi c\theta}{m} - g \\ \frac{(b(w_2^2 - w_4^2)cos|\theta_s|)l - \dot{I}_{xx}\dot{\phi}}{I_{xx}} \\ \frac{(b(w_3^2 - w_1^2)cos|\phi_s|)l - \dot{I}_{yy}\dot{\theta}}{I_{yy}} \\ \frac{d[(w_3^2 + w_1^2)cos|\phi_s| - (w_3^2 + w_1^2)cos|\theta_s|] - \dot{I}_{zz}\dot{\psi}}{I_{zz}} \end{pmatrix} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} \tag{15}$$

http://www.imavs.org/

Since the robot is at near hover condition, $\dot{I}_{xx}\phi$, $\dot{I}_{yy}\theta$ and $\dot{I}_{zz}\psi$ are equal to zero.

Clearly Eq.s (10 & 15) show that the generation of a yaw moment is unavoidable if the vector thrust is active. However, these equations of motion do not consider any input from the autopilot (ie assuming autopilot is inactive).

3.1 Plus-configuration vector thrust numerical simulations

Fig. 5 shows the MAV with the respective coordinate system. It is assumed that one desires a translation in y-direction. In order to carry out this motion without a change in roll angle, two servos attached to the motors 1&3 (Fig. 5 Right) must be active (in this case servos are set to be 5 degrees at the beginning). All motors rotate at a constant speed.

The simulation results displayed in Fig. 6 show that creation of a yaw moment is pos-sible while the vector thrust is active. No autopilot is involved. The flight control sys-tem does have heading (yaw) angle lock capabilities in this case. Fig. 7 shows the simulation results when both autopilot and vector thrust are active. The global translation is now limited to y and z (z is altitude), and x has been completely re-moved by locking the heading angle.

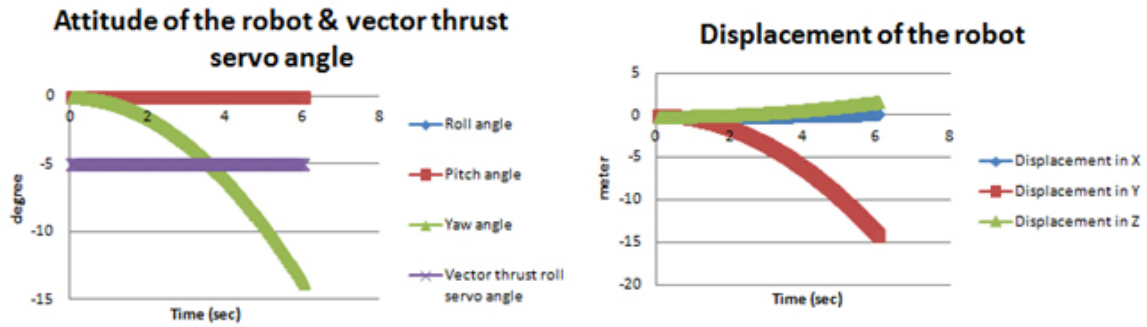


Figure 6: **Left**: Attitude change of robot when vector thrust is applied. **Note**: no change in roll and pitch, but yaw change is undesirable (cause of symmetric rotation of motor2&4 (Fig. 3.Right) **Right**: Global translation of robot. Note: y-direction is desirable, and x-direction is causing by unde-sired yaw.

4 X-configuration for vector thrust modeling

The main reason for the plus-configuration frame robot to shift its heading every time vector thrust becomes active is because asymmetry of the vector thrust servo and rotor rotation direction. However, this problem will not occur when using an x- configuration robot. For application of vector thrust, all motors rotate partially also around the x-axis to allow a momentum around the y-axis and rotate around y-axis to allow a momentum around the x-axis. All equations except for Eq.s (10), (14) and

http://www.imavs.org/

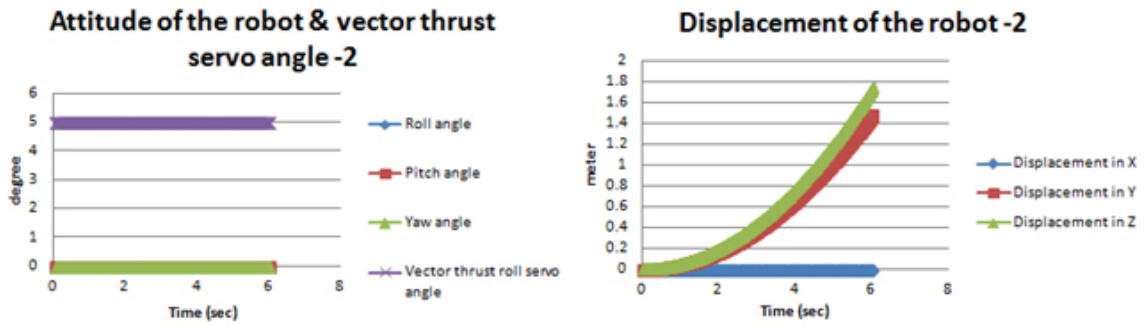


Figure 7: **Left**: Attitude change of robot when vector thrust is applied. **Right**: Global translation of robot. Note: no any undesirable translation in x-direction.

(15) are valid (since the motor numbering and orientation change; see Fig. 8). The modified equations are Eq.s (16), (17) and (18) respectively.

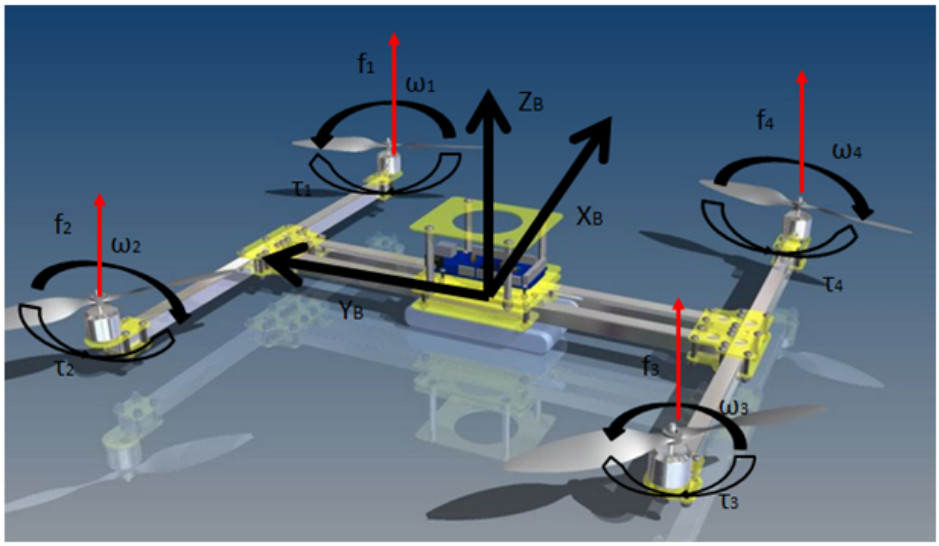


Figure 8: Shows force diagram of X-frame setup Quad-copter. Note: here we use H-frame because of simplicity for installation of vector thrust servo.[17].

$$F_x = -b(w_1^2 + w_2^2 + w_3^2 + w_4^2)sin\theta_s$$

$$F_x = b(w_1^2 + w_2^2 + w_3^2 + w_4^2)sin\phi_s$$

$$F_z = b(w_1^2 + w_2^2 + w_3^2 + w_4^2)\ cos|\theta_s|\ cos|\phi_s| \tag{16}$$

http://www.imavs.org/

$$\tau_\phi = (F_{1Z} + F_{2Z} - F_{3Z} - F_{4Z})lcos45 + (\tau_1 + \tau_2 + \tau_3 + \tau_4)sin\theta_s$$

$$\tau_\theta = (F_{1Z} + F_{2Z} - F_{3Z} - F_{4Z})lcos45 + (\tau_1 + \tau_2 + \tau_3 + \tau_4)sin\phi_s$$

$$\tau_\psi = (\tau_1 + \tau_2 + \tau_3 + \tau_4)\ cos|\phi_s|\ cos|\theta_s| \tag{17}$$

$$\ddot{q} = \begin{pmatrix} \frac{F_x c\theta c\psi + F_y c\theta s\psi - F_z s\theta}{m} \\ \frac{F_x(s\phi s\theta c\psi - c\phi s\psi) + F_y(s\phi s\theta s\psi - c\phi c\psi) - F_z s\phi c\theta}{m} \\ \frac{F_x(c\phi s\theta c\psi - s\phi s\psi) + F_y(c\phi s\theta s\psi - s\phi c\psi) - F_z c\phi c\theta}{m} - g \\ \frac{(b(w_1^2 + w_2^2 - w_3^2 - w_4^2)cos\theta_s cos\phi_s)lcos45 - d(w_2^2 + w_4^2 - w_1^2 - w_3^2)sin\theta_s - \dot{I}_{xx}\dot{\phi}}{I_{xx}} \\ \frac{(b(w_3^2 + w_2^2 - w_1^2 - w_4^2)cos\theta_s cos\phi_s|)lcos45 - d(w_2^2 + w_4^2 - w_1^2 - w_3^2)sin\phi_s - \dot{I}_{yy}\dot{\theta}}{I_{yy}} \\ \frac{d(w_1^2 + w_2^2 + w_3^2 + w_4^2)cos\theta_s cos\phi_s - \dot{I}_{zz}\dot{\psi}}{I_{zz}} \end{pmatrix} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} \tag{18}$$

where again, near hover condition, therefore $\dot{I}_{xx}\dot{\phi}$, $\dot{I}_{yy}\dot{\theta}$ and $\dot{I}_{zz}\dot{\psi}$ are equal to zero.

4.1 X-configuration vector thrust numerical simulation

Using Eq. (18) and assuming all 4 motors to rotate at the same speed the vector thrust roll servo angle is set to 5X and the vector thrust pitch servo angle to 0X. In this case the autopilot is inactive, thus no control for heading lock is used.

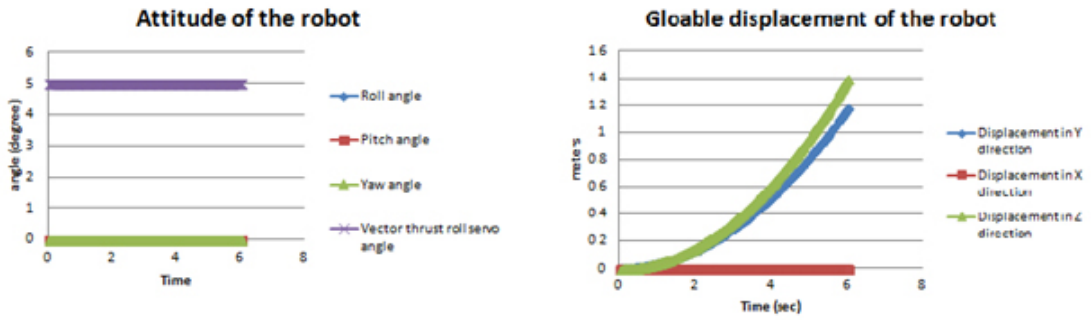


Figure 9: **Left**: Numerical simulation results of robot attitude when vector thrust roll servo is active at 5 degree. (**Note**: no autopilot control). **Right**: global displacement of the robot. **Note**: no undesired x displacement due to no yaw angle created.

Numerical simulation results displayed in Fig. 9 show that the performance of vector thrust using the x-configuration is much better (stable, without any input of autopilot) than the plus-configuration. However, the only problem is the actual design of an x-configuration vector thrust quadcopter as it may require 6 servos instead of 4 when the specific H-frame will be used as shown in Fig. 8.

http://www.imavs.org/

http://www.imavs.org/

5 Hardware realization and test flight

In order to confirm the numerical simulation, a simple cross-framed quadcopter was built as shown in Fig. 10 which could be flown in the plus- as well as the x-configuration. Flight tests were performed with the different configurations where videos have been taken from which at least two frames have been selected from the x-configuration flight and are shown in Fig. 11. When comparing the two pictures it can be seen that the vehicle stays continuously horizontal with no significant yaw being observed irrespective of the vehicle hovering or moving. This enhancement in attenuation therefore significantly helps to reduce the 'noise' with respect to the series of images taken.



Figure 10: **Left**: overview of the vector thrust Quad-copter. **Right**: note how the motor 1 and 3 tilt.



Figure 11: **Left**: hovering and vector thrust is activated. **Right**: 2 seconds after activation. Note: the quad moved in Y-axis without any attitude change.

6 Conclusion and future work

It has been shown that vector thrust propulsion of a multi-copter aerial vehicle does have a significant advantage on enhancing the attitude of the vehicle and hence the quality of any images taken from that vehicle with any type of sensing system. The analytical approach proposed can be programmed and implemented into the flight control of the aerial vehicle and first experiments performed indeed demonstrate the enhancement of the vehicle's attitude performance. As to the possible control approaches the x-configuration is definitely superior when compared to the plus-configuration. With the enhancement in attitude achieved it will now be the next steps to equip the aerial vehicle with the sensing system (i.e. a camera) and to observe up to which extent the gains are in terms of 'noise' obtained from the redundant images taken. There is also a drawback from the vector thrust application which can be seen from Fig. 7. With increasing time there is an increase in speed which results

in acceleration and which will theoretically lead to a gradual reduction of image quality. However how serious this effect is in reality needs to be proven by experiment and in the worst case this could be compensated through an additional velocity control. This concept has applied to the European Patent Office[18].

References

[1] C.-H. Kuo, A. Leber, C.-M. Kuo, C. Boller, C. Eschmann and J. Kurz. Unmanned robot system for Structure health monitoring and Non-Destructive Building Inspection, current technologies overview and future improvements; Proc. of the Internat. Workshop on SHM, Stanford Univ., Stanford/CA, USA, September 2013

[2] Golabi K, Thompson P, Hyman W. Pontis. A network optimization system for bridge improvements and maintenance Technical manual. Publication number FHWA-SA-94-031: US Department of Transportation, Federal Highway Administration; 1993.

[3] Brecher A.. Infrastructure: a national priority. Soc Women Eng 1995; 4(16):14-16.

[4] Roberts E, Shepard R.. Bridge management for the 21st century. Transport Res Rec 2000;1696:197-203.

[5] Bridge Inspection Robot Development Interface (BIRDI). Development of high-tech automatic robot system for bridge inspection and monitoring, Technical report for Ministry of Construction and Transportation (in Korean), 2007.

[6] Federal Emergency Management Agency (FEMA). Module 1C structural engineering, structural collapse technician course - student manual 8 `http://www.fema.gov/emergency/usr/sctc.shtm20098`(last visit Dec, 2009)

[7] Aldunate R., S.F. Ochoa, F. Pena-Mora, M. Nussbaum. Robust mobile ad hoc space for collaboration to support disaster relief efforts involving critical physical infrastructure, Journal of Computing in Civil Engineering 20 (1) (2006) 13-27.

[8] Federal Highway Administration (FHWA). Bridge Inspections Training Manual, July 1991.

[9] Bridge Maintenance Training Manual. US Federal Highway Administration, FHWAHI-94-034, Prepared by Wilbur Smith Associates, 1992.

[10] NJDOT, Bridge Inspection Work Zone. Setup Guide, 2009. `http://www.state.nj.us/`

[11] Shibata T. and A. Shibata. Summary report of research and study on robot systems for maintenance of highways and bridges, Robot, vol. 118, JARA, Tokyo, Japan, Sep. 1997, pp. 41-51.

[12] Product Catalog. Paxton-Mitchell Snooper? Series 140 `http://www.paxtonmitchell.com`

[13] Oh J.-K., A.-Y. Lee, S.M. Oh, Y. Choi, B.-J. Yi, H.W. Yang. Design and control of bridge inspection robot system, IEEE Int. Conf. on Mechatronics and Automation, Aug. 2007, pp. 3634-3639.

[14] Luukkonen T. Modelling and control of quadcopter; School of Science. Independent research project in applied mathematics Espoo, August 22, 2011. Aalto University.

[15] Group 10833. "Modelling and Control of Autonomous Quad-Rotor". 2nd Semester Project of the Intelligent Autonomous Systems Master Programme Faculty of Engineering, Science and Medicine. University of Aalborg, Denmark June 2010.

[16] Naidoo Y., R. Stopforth and G. Bright. Quad-Rotor Unmanned Aerial Vehicle Helicopter Modelling & Control; Intech open access publisher. Aug 2011

[17] X-configuration in H-Frame `http://aeroquad.com/showthread.php?4560-My-new-H-frame-design/page2`(last visit Nov, 2012)

[18] European Patent Office, Submission number: 2288267, Application: EP13183347.7, Reference Code: F54634-EP

http://www.imavs.org/

Technical Session 4

Vision

Chairmain : Guido De Croon

- Kirill Shilov, Vladimir Afanasyev and Pavel Samsonov. ***Vision-based Navigation Solution for Autonomous Indoor Obstacle Avoidance Flight***
- Andrew Nolan, Daniel Serrano, Aura Hernandez, Daniel Ponsa and Antonio Lopez. ***Obstacle mapping module for quadrotors on outdoor Search and Rescue operations***
- Jose Luis Sanchez-Lopez, Alberto Moreno, Jesus Pestana and Pascual Campoy. ***Visual Quadrotor Swarm for IMAV 2013 Indoor Competition***
- Johanna Matthaei, Thomas Krüger, Stefan Nowak and Ulf Bestmann. ***Swarm Exploration of Unknown Areas on Mars Using SLAM***

http://www.imavs.org/

Vision-Based Navigation Solution for Autonomous Indoor Obstacle Avoidance Flight

Kirill E. Shilov[1], Vladimir V. Afanasyev[2] and Pavel A. Samsonov[3]

[1] Moscow Institute of Physics and Technology (MIPT), Zhukovsky, Russia
aviaks.kirill@gmail.com
[2] Moscow State University, Moscow, Russia
vvafanasjev@mail.ru
[3] Moscow State University, Moscow, Russia
aspsnaky@gmail.com

http://www.imavs.org/

Abstract

This paper describes scientific and technical approaches as well as the results obtained during the research and flight tests aimed at enhancing MAV autonomy by the means of computer vision sensing. An inspiration for the research and development of video-based navigation system was given by the task to autonomously overcome the distance in a corridor containing randomly set obstacles. Being located indoor in GPS-denied space the corridor size is slightly more than a vehicle size. Medium-sized hexacopter was chosen as an airframe. Onboard electronics consists of two parts: flight control module and video processing module. Flight control module is based on author's custom SmartAP autopilot (presented at IMAV2012) and provides low-level services such as hexacopter stabilization and point-to-point flight routing. Video processing module is responsible for position estimation, obstacles determination and further flight trajectory calculation. Flight control module guides the MAV in order to follow the desirable path generated by video processing system. Technical details of the setup and computational solutions are presented in the paper.

1 Introduction

The typical mission to be completed consists of start zone, landing zone, two corridors and randomly set obstacle. MAV should overcome the distance without pilot control, including autonomous take off, randomly set obstacles detection and avoidance, autonomous landing.

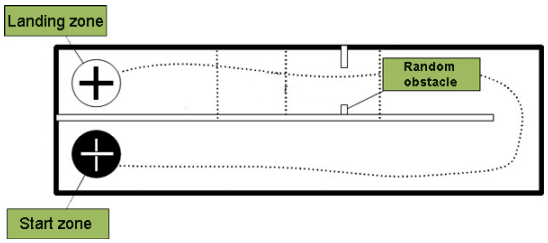


Figure 1: Scheme of the mission to be completed

Start zone and landing zone are marked with the big circles of black and white color respectively with the plus signs. The diameter of the circles is 3 meters. Corridor length is 50 meters, width is 8 meters. Corridor walls are painted in light-blue color.



Figure 2: Corridor with landing zone circle

2 General approach

In order to perform the flight described MAV must have both Flight Control System (FCS) which is responsible for MAV stabilization and control and Computer Vision System (CVS) for obstacles detection and path planning.
SmartAP autopilot [1] was chosen as FCS module. CVS module is based on laptop's motherboard instead of single-board computer because computational resources are critical for the task and can be provided only by high-end CPU and GPU.

3 FCS module

Flight control system hardware contains MCU, sensors and external ports for peripheral connections. This allows providing sustainable navigation solution.
The key features of the FCS are:

- Powerful microcontroller 72MHz ARM Cortex M3 MCU [2]
- 9-Degrees Of Freedom Inertial Measurement Unit [3]
  - 3-axis accelerometer
  - 3-axis gyroscope
  - 3-axis magnetometer [4]
- Pressure sensor for altitude measurement [5]
- Differential pressure sensor for air speed measurement
- GPS receiver for global position determination [6]
- Wireless data channel for two-way telemetry
- SD-card for in-flight data logging
- USB port for PC/Laptop connection for firmware uploading, debugging and testing routines
- PWM inputs/outputs to read signals from standard RC receiver and send them to motors/servos
- Several ADC channels for battery voltage/amperage monitoring
- Analog inputs for ultrasonic rangefinders connection

http://www.imavs.org/

The FCS software architecture consists of several levels: from the lowest - STM microcontroller libraries to the highest flight control logic. CMSIS Library from STM makes possible to create higher software hardware-independent. The second level is STM Library which provides high-level functions for microcontroller peripheral communications. Next level includes drivers for sensors readings and actuators control. The highest level consists of functions responsible for stabilization, navigation, flight control and digital signal filtering where its necessary. Additionally, the highest level includes GCS interface functions based on MAVLink communication protocol library. The entire software is run on FreeRTOS (Real Time Operating System) developed especially for embedded applications which provides functionality for effective tasks and resources control being crucial for such devices. Every function belongs to its priority group corresponding to its importance for safety guideline. Therefore, FCS block diagram can be presented in the following way:

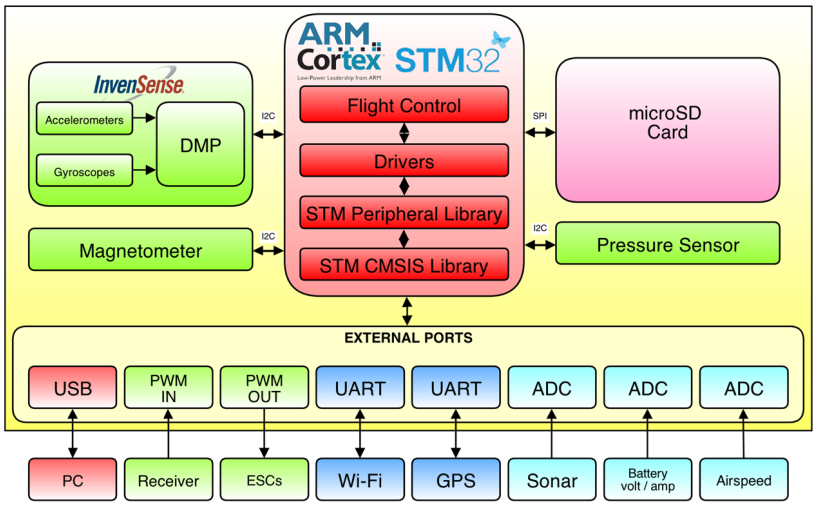


Figure 3: FCS block diagram

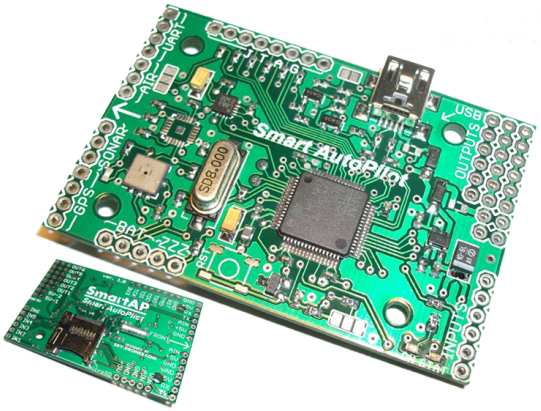


Figure 4: Assembled FCS hardware

http://www.imavs.org/

http://www.imavs.org/

4 CVS Module

On-board video-processing hardware contains:

- Laptop motherboard (Asus N46VZ)
- High-end mobile CPU Intel Core i7 3670QM
  Middle-end GPU NVidia GT650M
  8 GB of DDR3 system memory and 2 GB of GDDR5 video memory
- 3 industrial high-speed USB 3.0 cameras (Ximea MQ013CG-E2)
  1.3 MPixel resolution
  60 FPS

CPU and GPU computational resources are used for high-speed 3D map generation. Cameras with fixed positions form a number of stereograms. This allows scene depth calculation.
Flight control and video processing modules are connected via USB. Flight control module sends inertial data information and receives back adjusted position, velocity and desired position for the following iteration. Additionally, airframe is equipped with 4 ultrasonic rangefinders. One of them is facing downwards (for altitude estimation), the other three are facing straight forward and are responsible for collision avoidance in case of noisy video gathered data.
The video processing module receives image frames and sensor data. Sensor data is used as initial approximation for the vehicle position refinement. First, the frames are preprocessed in order to get points for effective local 3D reconstruction. Each frame gives a small number of reliable 3D object points that supplement and refine the whole 3D map. Restricted and permitted areas are marked on this map and are renewed during the flight. The desired trajectory to the destination is calculated using the map and the nearest target point is transmitted to the flight control module.
On this basis, flight control module and video processing module form sustainable electronic platform allowing MAV to fly indoor autonomously.

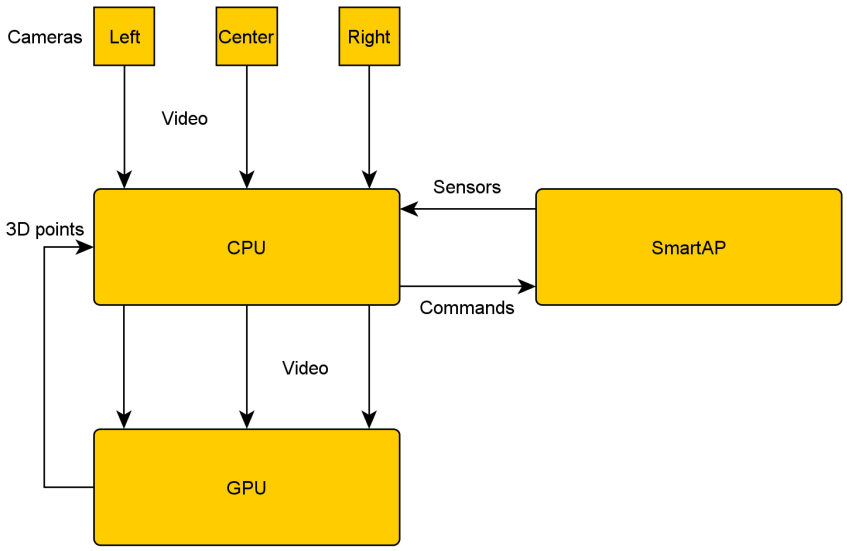


Figure 5: CVS Hardware

http://www.imavs.org/

5 Airframe

Since the laptop motherboard is quite heavy it was decided to use hexacopter airframe instead of traditional quadcopter to allow higher carrying capacity. The motherboard is installed at the top of the hexacopter. FCS is mounted on the motherboard. Hexacopter is assembled to fly in + mode meaning that the one arm is facing forward. Three video cameras are mounted to the end of the arms. Total weight of the airframe is about 2700g.

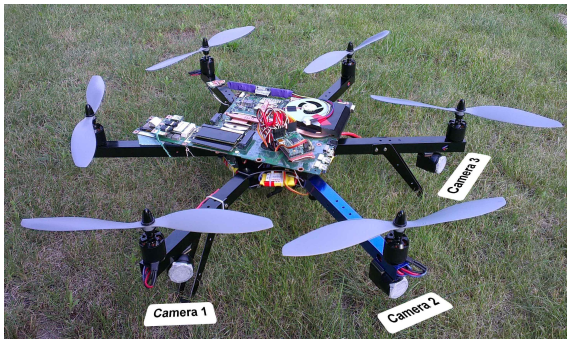


Figure 6: Hexacopter airframe

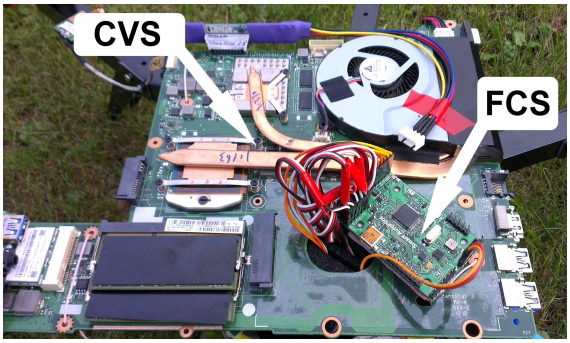


Figure 7: Close-up of CVS and FCS

6 Video processing

The emphasis is done on 3 cameras system mostly because of the laptop motherboard has only 3 USB 3.0 ports. Though for the algorithm presented even 2 cameras are sufficiently, 3 cameras will provide wider angle of view.

The video processing system is based on handling several high-fps video streams. Each of video streams has outputs 50 FPS. So, motherboard is intended to process 150 frames per second. Being uncompressed this data stream is approximately 600 MB/s. CPU is able to perform only several operations by pixel which is not enough That's why GPU instead of CPU is used.

To provide high MAV flight speed high-speed megapixel cameras were chosen. The higher frame rate allows less lookup size during each next frame processing in case of inaccurate position prediction or its absence.

Some technical parameters of the camera are presented below:

- Physical frame rate: 60 FPS, 50 FPS actual
- Resolution: 1280x1024
- 8 bit RGB color
- CMOS sensor with global shutter
- Lens: 6 mm CS, 90 deg. horizontal, 60 deg. vertical angle of view
- USB 3.0 interface

Camera specifications impose restrictions on the use of algorithms:

1. The high frame rate leads to low exposure time. If the frame rate is 50 FPS, exposure time for each frame must be less than 20 ms. On a sunny day 0.5 ms exposure time is required, on a cloudy day exposure should be 5-10 ms. The indoor artificial lighting is the limit for such cameras at a high frame rates. The images become dark, though, image processing is still able to handle camera data.

2. Asynchronous mode for higher frame rates must be applied. There is a possibility to synchronize the cameras in hardware, but this leads to a situation when one camera waits for another each frame, so overall frame rate degrades. It was decided that the higher frame rate is more important for the task rather than simultaneity of the frames.

6.1 Preparation part

The algorithm initializes with camera positions and rotation calibration. By now, this part is performed manually. Recalibration should been done is case of cameras were moved. After that, distortion parameters acquisition is performed. The automatic fitting of Broun model using a checkerboard is performed for every lens type. Normally, it is made only once for each camera.
Finally, the exposure adjustment algorithm starts. The exposure value is selected to provide normal average brightness in the frame. Firstly, optimal exposure is measured for each camera. After that the medium exposure is calculated and set for all cameras to have the same brightness for all images. At the moment, only static exposure calculation is implemented. Having been set once, it doesn‘t vary a lot during the flight. However, dynamic exposure adjustment is also planned to be done.

6.2 Image processing pipeline

Images are acquired from each camera in separate threads. On the maximum FPS each camera thread uses almost 100% load of a single CPU core. The most of processor time is utilized in camera driver. So, we have only one core left for all remaining tasks. Then the image is passed from CPU to GPU through PCI-E 16x 2.0 interface. The CPU task is to get the image from camera driver and pass it to GPU. This is a necessary minimum that should be done with such huge data. In this manner the load from CPU put off as much as it is possible.
The distortion in the image is corrected with pre-built correction map that contains new coordinates of each pixel. The white balance is corrected right after distortion. Color correction parameters are set only once for each flight.
After that the image is adaptively binarized to find the cross in the image. All image is processed by Gaussian blur and subtracted from initial image. The difference image is then binarized using a static threshold.
The feature points are allocated in the binary image. A fast feature detection algorithm is used. Then the feature points 3D positions are reconstructed using the correspondence lookup on camera pairs forming stereograms. Features are also matched on sequential images from the same camera to get 3D point correspondence and calculate the new MAV position. Due to the fact that images are shot asynchronously by cameras and MAV has non-zero speed we get a variable addition to a constant stereogram configuration. Variations in stereogram parameters may lead to significant errors in stereo reconstruction. To minimize these errors stereogram camera positions and orientation are corrected using extrapolation of the previous MAV dynamics as well as inertial sensor readings.

7 MAV position estimation

The key idea of position estimation algorithm is to combine data from all sensors. All sensors provide different data with different accuracy and different types of errors.

http://www.imavs.org/

Sensor	Accuracy factors	Accuracy
Video	Camera calibration accuracy Distortion correction quality Processing time restrictions	Up to 1 cm at 5 m distance (depth map)
Gyroscopes	Vibration	0.001 deg/s
Accelerometers	Airframe vibrations Update frequency Filtration technique	0.01-0.1g
Ultrasonic rangefinders	Sound interference Motors air flow Surface type	Up to 1 cm at 1-2 m distance

Every type of sensor has its own failure conditions and reliability. Normally, all sensors give correct data, but one of the sensors can give data with errors. Every new piece of data from some sensor is checked for consistency. If current sensor does not form a consensus with other sensors, it is marked as unreliable and is not used to form the final position estimation.
The specific properties of every sensor are also taken into account to use its data more efficiently. Video as the most precise and reliable sensor is used to make check points in MAV trajectory.

8 Control

FCS is responsible for low-level functions (stabilization, point-to-point flight) while CVS is responsible for obstacle detection and avoidance and path-planning.

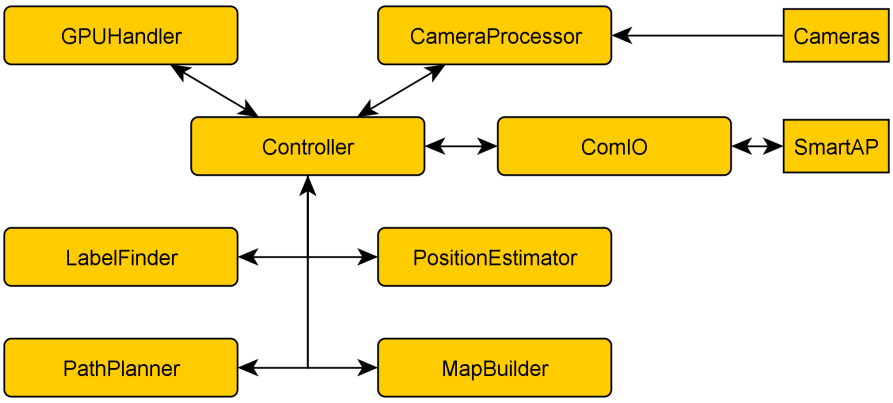


Figure 8: CVS Software

Basically, it can be subdivided into three groups:

- Targeting module
- Path planning module
- Interface with FCS

http://www.imavs.org/

FCS path-following is done using PI-PI controller which diagram can be presented as follows:

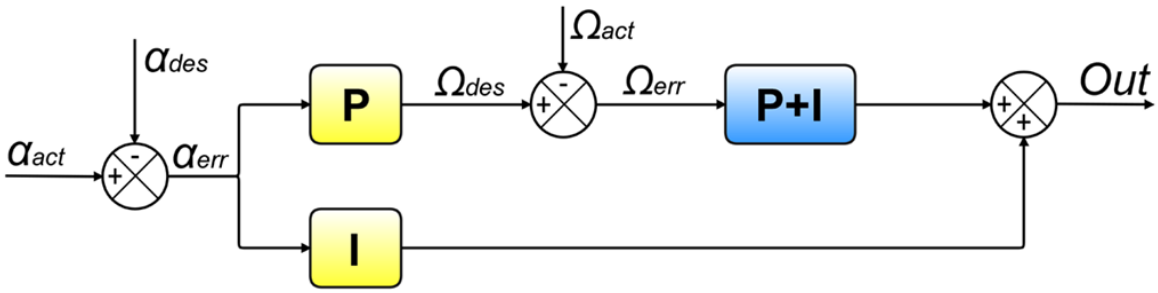


Figure 9: PI-PI Control loop for FCS

9 Conclusion

By the moment airframe had been already assembled and performed several flight tests. Computer vision system algorithms have been validated in simulation and confirmed the ideas described in the paper. Hexacopter is able to demonstrate autonomous take off and landing with position hold based on the information received from computer vision module. Fully autonomous flight in GPS-denied environments and on the fly 3D map generation are the subjects of further research in this project.

10 Reference

[1] MAV Autopilot for Commercial and Research Purposes, Kirill Shilov, Grigory Lazurin, IMAV 2012, Braunschweig, Germany, 3-6 July 2012

[2] High-density performance line ARM-based 32-bit MCU, STMicroelectronics, http://st.com/

[3] MPU-6000 and MPU-6050 Product Specification Revision 3.2, Invensense, http://invensense.com/

[4] 3-Axis Digital Compass IC HMC5883, Honeywell, http://honeywell.com/

[5] BMP085, Digital, barometric pressure sensor, BOSCH, http://www.bosch-sensortec.com

[6] UBlox LEA-6, u-blox 6 GPS Modules Data Sheet, http://www.u-blox.com/

http://www.imavs.org/

Obstacle mapping module for quadrotors on outdoor Search and Rescue operations

Andrew Nolan[1], Daniel Serrano[2], Aura Hernandez Sabaté[1], Daniel Ponsa Mussarra[1] and Antonio M. López Peña[1]

[1] Centre de Visió per Computador, Universitat Autònoma de Barcelona, Bellaterra, Spain
andrewpeternolan@gmail.com
[2] Ascamm Private Foundation, Barcelona Cerdanyola del Vallès, Spain
dserrano@ascamm.com

http://www.imavs.org/

Abstract

Obstacle avoidance remains a challenging task for Micro Aerial Vehicles (MAV), due to their limited payload capacity to carry advanced sensors. Unlike larger vehicles, MAV can only carry light weight sensors, for instance a camera, which is our main assumption in this work. We explore passive monocular depth estimation and propose a novel method Position Aided Depth Estimation (PADE). We analyse PADE performance and compare it against the extensively used Time To Collision (TTC). We evaluate the accuracy, robustness to noise and speed of three Optical Flow (OF) techniques, combined with both depth estimation methods. Our results show PADE is more accurate than TTC at depths between 0-12 meters and is less sensitive to noise. Our findings highlight the potential application of PADE for MAV to perform safe autonomous navigation in unknown and unstructured environments.

1 Introduction

The use of Unmanned Aerial Vehicles (UAV) has become an important part of search and rescue operations in recent years. UAV can be used to assist human teams with multiple operations; from exploring a disaster area, to dealing with the difficult task of finding human survivors. The European Union project, Integrated Components for Assisted Rescue and Unmanned Search operations (ICARUS EU-FP7) [5] aims to improve unmanned air vehicles to integrate them into existing emergency service systems. As part of the ICARUS project, Aerospace technology centre of Fundació ASCAMM (Spain), is developing a small quadrotor platform to aid in Urban Search and Rescue (USAR) operations. ASCAMM's quadrotor is considered a Micro Aerial Vehicle (MAV) due to its size and limited payload capacity. USAR scenarios require MAV to perform safe autonomous navigation in unknown and unstructured outdoor environments. In order to autonomously navigate MAV needs to plan paths that avoid collisions with any obstacles.

Path planning can be described in two steps; firstly, to use sensors to generate a description of the environment and identify any potential obstacles. Then, to find a feasible path between two points using the environment description that does not result in a collision [27]. A description of the environment can be achieved through depth estimation using a variety of active and passive sensors. Active depth sensing uses physical sensors such as laser or structured light patterns [25] to obtain depth information from natural scenes [11]. For large quadrotors, collision avoidance has been solved using active sensors such as laser range finders [22] or depth cameras [10]. However, on MAV with limited payloads, active sensors remain prohibitive because they are power consuming and heavy. In

contrast to active sensors, passive visual sensors such as cameras present a light weight option for collision avoidance on MAV.

Passive sensors such as stereo or monocular cameras estimate depth using stereopsis methods that mimic the human visual system by calculating depth from two slightly different viewpoints of the same scene [20]. Stereo cameras use distance between each viewpoint to recover depth by measuring the relative displacement of an object on two image planes [14]. Similarly, depth is recovered from a monocular camera by estimating motion on the image plane using Optical Flow (OF) methods and measuring camera translation and rotation [18, 27]. Monocular cameras are best suited for visual navigation on MAV due to their light weight and low power requirements.

The primary focus of our work is the investigation of monocular depth estimation techniques using stereopsis methods. For robotic (monocular) depth estimation, the most extensively used technique is called Time To Collision (TTC) [23, 17, 4]. TTC estimates, for each pixel, the number of frames until a collision occurs. The TTC algorithm starts by removing the effects of camera rotation from the flow field by either estimating camera rotation or using measurements obtained from an Inertial Measurement Unit (IMU). Next, the location of the flow field's Focus of Expansion (FOE) is estimated and used to calculate the time to collision for each flow vector. Using the vehicle's translational velocity, each TTC value is converted to a depth estimate [23, 28]. TTC relies on an accurate FOE estimation, which in practice is not a trivial task for arbitrary camera motion and especially difficult under noisy conditions [2].

In our work, we investigate possible methods for obtaining a description of the environment using depth estimation in the context of a quadrotor MAV. Many OF based depth estimation techniques have been presented for MAV [16, 26] operating indoors and outdoors, although few have incorporated camera position to improve depth estimation. Our state of the art review has shown a lack of research combining OF techniques with tightly coupled Global Positioning System (GPS) and Inertial Navigation System (INS) measurements for depth estimation. Fusing GPS/INS measurements with an Extended Kalman filter (EKF) [6] improves the accuracy of relative position estimates [1]. In our work we propose a novel Position Aided Depth Estimation (PADE) method and compare its accuracy and computational performance against TTC. We also evaluate the robustness of both methods under noise.

http://www.imavs.org/

2 Methodology

Our general processing pipeline for monocular depth estimation, shown in Figure 1, is constructed of two processing stages: Data Acquisition and PADE. The Data Acquisition stage uses two consecutive images to calculate optical flow and also calculates the relative camera translation and rotation from absolute position and orientation measurements. The PADE stage uses OF and camera rotation and translation to calculate a Depth Image.

Motion observed on the image plane is known as Image Flow. Image Flow is the result of the projection of an object's 3D velocity on the image plane [8] generated by objects in the scene moving or through camera movement. To approximate image flow, OF methods detect motion between two consecutive images for each pixel, described by a 2D vector $[uv]^T$. If we assume the scene is static and the observed motion is generated by camera translation and rotation, the computed OF can be used to reconstruct 3D structure of a scene through flow divergence techniques [18].

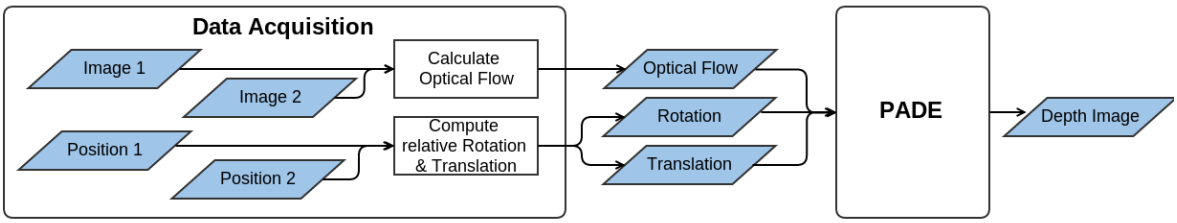


Figure 1: Monocular depth estimation pipeline

2.1 Optical Flow

The relationship between camera motion (translation and rotation) and OF can be defined as the velocity of a 3D point in space with its corresponding velocity on the image plane, shown in the equation:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{Z}\mathbf{A}\mathbf{t} + \mathbf{B}\omega \tag{1}$$

Matrices $\mathbf{A}$ and $\mathbf{B}$ are constructed using the camera focal length f and pixel coordinates x and y.

$$\mathbf{A} = \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -\frac{xy}{f} & f + \frac{x^2}{f} & -y \\ -f - \frac{y^2}{f} & \frac{xy}{f} & x \end{bmatrix}$$

Camera translation and rotation are defined by vectors $\mathbf{t} = [t_t, t_y, t_z]^T$ and $\omega = [r_x, r_y, r_z]^T$ and are expressed relative to the first image frame. In our work we use the convention t_z as forward, t_x as right, t_y as down for camera translation and rotation is represented as using typical avionics notation r_x for roll, r_y for pitch, r_z for yaw.

Flow divergence generated by pure camera translation indicates the 3D location of stationary objects in a scene. Flow divergence methods assume that an object increases in size on the image plane as the distance to the object decreases, thus flow magnitude is inversely proportional to the depth of a point to the camera. Using the spatial arrangement cues generated by the relative motion between a scene and a viewer we can reconstruct a model of the observed scene.

2.2 Compute Relative Translation and Rotation

Equation (1) defines OF as the result of objects 3D velocity on the image plane [8]. If we assume that the scene is static, then we can suppose that motion observed on the image plane is the result of camera translation and rotation. The camera's relative translation $\mathbf{t} = [t_t, t_y, t_z]^T$ is the position displacement between the position p at time t and the position at time $t + 1$, shown in the following equation:

$$\mathbf{t} = \mathbf{p}_{t+1} - \mathbf{p}_t \tag{2}$$

Similarly, relative camera rotation $\omega = [r_x, r_y, r_z]^T$ is the difference between the orientation o at time t and the orientation at time $t + 1$, shown in the following equation:

$$\omega = \mathbf{o}_{t+1} - \mathbf{o}_t \tag{3}$$

http://www.imavs.org/

2.3 Position Aided Depth Estimation (PADE)

Monocular depth estimation consists of reconstructing a 3D scene using depth cues extracted from consecutive image frames after camera translation and rotation. The relative depth image of the environment can be computed from OF and position information. Our PADE method uses relative camera position to improve OF depth estimation. To estimate depth, it is not necessary to compute the FOE since we already have the camera's translation and rotation $[\mathbf{t}, \omega]$. Figure 2.3 shows our PADE pipeline as two steps: Average Optical Flow and Calculate Depth Image.

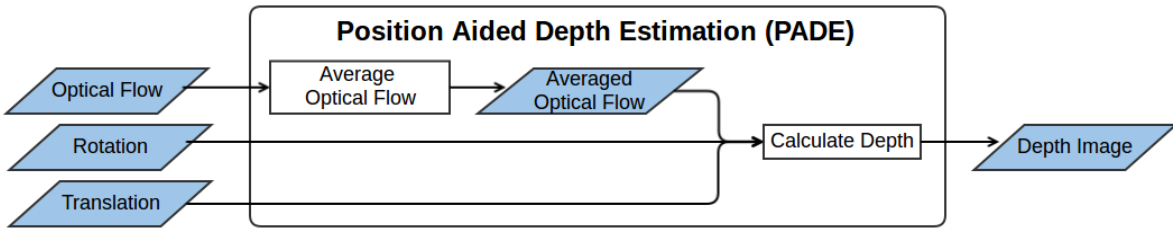


Figure 2: Position Aided Depth Estimation (PADE) pipeline

2.3.1 Average Optical Flow

Computing a dense depth image from a dense OF field is computationally expensive because a depth value is computed for each flow vector. We aim to improve depth estimation performance by reducing the number of calculations. We calculate the average of all flow vectors within a $n \times n$ window and use it to compute a single depth value for the whole window. By reducing the number of depth calculations we can improve computational performance by $(n \times n) - 1$. Averaging OF also has the advantage of smoothing noisy vectors.

2.3.2 Calculate Depth

We isolate Z from equation (1) to compute depth using OF $[u, v]^T$, translation and rotation $[\mathbf{t}, \omega]$. By rearranging the terms in equation (1) we obtain the distance to a point Z, as follows:

$$Z = \mathbf{At} \cdot (\begin{bmatrix} u \\ v \end{bmatrix} - \mathbf{B}\omega)^{-1} \tag{4}$$

The camera's extrinsic parameters $\mathbf{t}$ and ω between image frames plus matrices $\mathbf{A}$ and $\mathbf{B}$ (defined in Section 2.1) allow us to solve for Z using an estimate of OF. Equation 4 requires translation $|\mathbf{t}| > 0$ for depth estimation.

3 Experiments

3.1 Measures

In this study, we have investigated multiple methodologies for obstacle mapping on small UAVs. Our primary research focus is the comparison of two monocular depth estimation techniques: TTC and PADE. To provide a quantitative assessment of both depth estimation techniques we used Robot

http://www.imavs.org/

Operating System (ROS) [21] with the robotic simulator Gazebo [12] to obtain ground truth depth images. We generated ten image sequences captured from a virtual quadrotor flying through a 3D scene. Our sequences, totalling 1038 frames at a resolution of 640×480, isolate many combinations of translation and rotation in order to explore the robustness of each depth estimation technique.

OF based solutions are typically a trade off between accuracy and speed. Our work focuses on evaluating methods with the potential of real-time speed at the expense of accuracy. We evaluate two dense global OF methods (CLG [3], Farnebäck [7]) and one local method (Lucas-Kanade [13]).

Absolute Depth Error (ADE)

We evaluate two depth estimation methods TTC and PADE combined with each OF method, using Absolute Depth Error (ADE). ADE is the difference between the estimated distance and the ground truth distance.

$$ADE = |Z_C - Z_T| \tag{5}$$

3.2 Results

In order to assess the accuracy at different depth ranges we have calculated ADE at 4 meter intervals (0-4m, 4-8m, 8-12m, 12-16m). Table 1 is divided into two blocks, TTC and PADE, and for each depth estimation method we evaluate the contribution of each OF method. For each 4 meter interval (columns), we report the average error and standard deviation. The minimum error for each depth range is highlighted in bold. Lucas-Kanade provides the lowest average error in the 0-8 meter range. TTC and PADE are inseparable for both Farnebäck and Lucas-Kanade, in the 0-4 meter range, although between 4-12 meters the difference between the two methods grows in favour of PADE.

	Absolute Depth Error (ADE)			
	0-4m	**4-8m**	**8-12m**	**12-16m**
TTC				
CLG	0.53 ±0.17	1.99 ±0.46	2.69 ±0.36	**2.15** ±0.47
Farnebäck	0.38 ±0.20	1.84 ±0.42	2.76 ±0.46	2.36 ±0.47
Lucas-Kanade	**0.37** ±0.18	1.76 ±0.36	2.53 ±0.37	2.41 ±0.42
PADE				
CLG	0.45 ±0.20	1.57 ±0.20	**2.47**±0.33	2.98±0.44
Farnebäck	0.40±0.16	1.53±0.28	2.54±0.32	3.23±0.47
Lucas-Kanade	0.39±0.15	**1.39**±0.28	2.51±0.35	3.47±0.45

Table 1: Absolute Depth Error for each distance range.

Position Measurement Noise Analysis

To analyse how dependent each depth estimation technique is on the accuracy of relative position measurements, we add noise to the translation, rotation and linear velocity measurements of up to 20%. If we consider a camera operating at 10 fps and a quadrotor with a translational velocity of 2 meters/second, an image will be captured approximately every 20 centimetres, producing translational errors of up to 4 cm in each direction. Figure 3 shows ADE for TTC and PADE using Lucas-Kanade with noise added to position measurements, PADE (green dotted), PADE with noise (green solid), TTC (red dotted) and TTC with noise (red solid). Lucas-Kanade results are representative of all OF methods, for simplicity they are the OF method shown in Figure 3. Consistent with our previous experiments, PADE (green dotted) has the lowest ADE between 0-12 meters and TTC (red dotted) is slightly better between 12-16 meters. With the addition of noise, PADE (green solid) ADE slightly increase, although it is stable from 0-16 meters. In contrast, TTC (red solid) ADE measurements

http://www.imavs.org/

increase by 10-30 centimetres between 0-12 meters and up to 65 centimetres between 12-16 meters. Our data suggests that PADE (green solid) is more robust to noise at 0-12 meters than TTC (red solid) by approximately 30%. TTC (red solid) is sensitive to noise because it is dependant on accurate linear velocity measurements to scale from frames into meters.

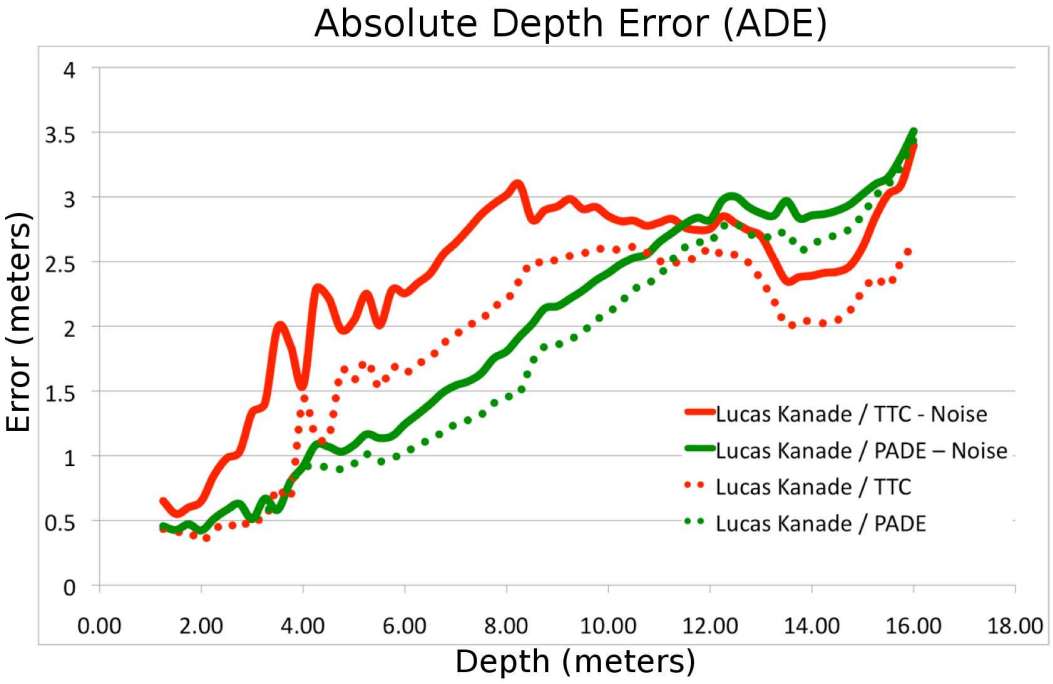


Figure 3: Absolute Depth Error - TTC and PADE with noise added.

Depth Estimation performance

To evaluate the computational speed of each step in our processing pipeline, Table 2 is divided into three main columns: OF, depth estimation and overall performance. We have evaluated the performance of each combination of OF and depth estimation. In both the OF and depth estimation sections we list the methods used and average processing time per frame in milliseconds. The final column (performance) shows the average processing time for the complete pipeline per frame. We have also included the approximate frames per second (FPS) measurement to indicate if each pipeline is appropriate for real time performance. The computational performance of OF methods vary substantially based on their implementations. Computations have been performed on a 2.5 GHz Intel Core processor executing C/C++ code.

Although Lucas-Kanade has out performed CLG and Farnebäck (see Table 1), its average processing time of 2800 milliseconds makes it inappropriate for real time performance. However, ADE measurements indicate that Farnebäck's accuracy is comparable to Lucas-Kanade in the range of 0-12 meters and has an average processing time of 100 milliseconds, 3% of Lucas-Kanade's. The fastest computational pipeline is Farnebäck combined with PADE, which achieves an average pipeline processing time of 136 milliseconds per frame or 7.35 fps.

http://www.imavs.org/

Optical Flow		Depth Estimation		Performance	
Method	Time(ms)	Method	Time(ms)	Total(ms)	FPS
CLG	880	TTC	190	1070	0.93
		PADE	39	919	1.09
Farnebäck	**100**	TTC	178	278	3.60
		PADE	**36**	**136**	**7.35**
Lucas-Kanade	2800	TTC	181	2981	0.34
		PADE	39	2839	0.35

Table 2: Optical flow / depth estimation density and processing time comparison.

4 Applications

4.1 Relative Positioning System

PADE requires accurate relative camera position and orientation to aid in depth estimation. Global Positioning System (GPS) currently provides position measurements with meter level accuracy. Operating on the micro scale compared to GPS there is Inertial Navigation System (INS). INS provides accurate dead reckoning, based on changes in linear and angular acceleration, although position uncertainty grows over time as errors accumulate. INS are typically implemented using an Inertial Measurement Unit (IMU) consisting of gyroscopes for angular measurements and accelerometers for linear acceleration. GPS and INS measurements are inherently noisy, although, by taking advantage of the strengths of both systems GPS and IMU signals can be fused together using an Extended Kalman Filter (EKF) to achieve greater precision [19]. GPS/INS can be combined by loose (integration at the position, velocity and/or attitude level) or tight integration (integration at the pseudorange, Doppler, or carrier phase level) [1] to correct measurements and even provide position estimates during short GPS outages.

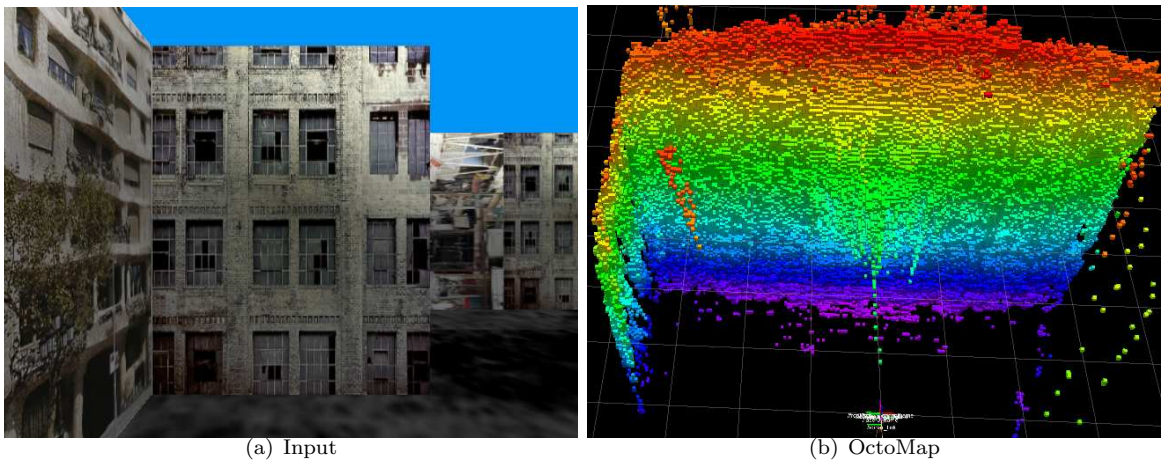
We propose using tightly coupled GPS/INS to improve the computing the camera's relative position and orientation measurements to aid in monocular depth estimation. If the observed scene is generally static, the displacement of pixels on the image plane is due to camera translation and rotation. Two consecutive frames can be treated as a stereo pair using the translation and rotation obtained from the GPS/INS as the camera baseline. The relative position and orientation between images depends on the frame rate of the camera and the UAVs speed. If we consider a camera operating at 10 fps and a UAV with a translational velocity of 2 meters/second, an image will be captured approximately every 20 centimetres. The accuracy of GPS/INS relative measurements for small position changes, and within short time periods is in the order of centimetres.

4.2 Occupancy Grids

Although it is possible to map obstacles using dense point clouds, it is more practical to create a coarse and fast representation of the surrounding scene by using an alternative representation, like an occupancy grid. An occupancy grid is a binary map representation of obstacles in the surrounding environment. Space is evenly divided into cells and each cell is determined to be occupied or empty. The simplicity of occupancy grids have made them the dominant spatial representation used in mobile robotics [24]. 2D occupancy grids are common for ground vehicle and 3D occupancy grids for air vehicles. Taking advantage of this mapping simplification we can use point clouds projected from depth images to update a probabilistic 3D occupancy map using a framework called OctoMap [9]. The

http://www.imavs.org/

occupancy grid when updated with a new point cloud takes into account the probabilistic uncertainty of noisy measurements, producing a low noise 3D obstacle map of the local environment. OctoMap uses a volumetric representation (voxel) with configurable resolutions. For our research we've used a voxel resolution of 10 centimetres. The occupancy grid is updated with reference to a global frame taking into account changes in the quadrotors position. The map's volumetric representation enables the quadrotor's trajectory planner to avoid collisions and ignore obstacles outside its planned path.



(a) Input (b) OctoMap

Figure 4: Image processing pipeline. Quadrotor flying towards a building.

5 Conclusions

In this work we explore monocular depth estimation techniques using stereopsis methods. We evaluate the extensively used Time To Collision (TTC) method against our novel Position Aided Depth Estimation (PADE) method and we compare both methods ability to measure distances up to 16 meters with camera translation and rotation. Our study includes an evaluation of the influence of OF methods on depth estimation and we evaluate three popular OF methods Lucas-Kanade, Farnebäck and Combined Local Global.

By adding noise to translation, rotation and velocity measurements, the difference between the two methods is more noticeable. Our study shows that TTC is sensitive to noise due to its reliance on linear velocity to scale its initial collision estimate from frames to meters. Small errors in the linear velocity estimation result in large depth errors. In contrast, the accuracy of PADE is only slightly reduced across all evaluated depth ranges, indicating that it is robust to noise.

As part of our evaluation of OF and depth estimation techniques we also evaluate the processing time for each step of the pipeline. Calculating motion using Lucas-Kanade produces the most accurate depth estimation. However, Lucas-Kanade's processing time makes it prohibitive for real time performance. Farnebäck on the other hand, provides comparable depth estimation results and requires only 3% of the time. The combination of Farnebäck with our PADE method provides a depth estimation pipeline with close to real time performance.

Our evaluation of depth estimation techniques in the context of MAV has shown that monocular depth estimation is improved when combined with relative camera position.

http://www.imavs.org/

6 Future work

Our initial results are encouraging, although future work will involve testing, evaluation and optimisation using real flight data before deployment on a quadrotor. Currently, the majority of image pipeline processing time is consumed by OF. However, using Graphic Processing Units (GPU) have been shown to compute OF at 25 fps [15], which could provide better pipeline throughput.

7 Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 285417. This work was supported by the Spanish projects TRA2011-29454-C03-01 and TIN2011-29494-C03-02.

http://www.imavs.org/

References

[1] Santiago Alban, D Akos, Stephen M Rock, and Demoz Gebre-Egziabher. Performance analysis and architectures for ins-aided gps tracking loops. *Institute of Navigation–NTM*, pages 611–622, 2003.

[2] Christof Born. Determining the focus of expansion by means of flow field projections. 1994.

[3] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[4] Jeffrey Byrne and Camillo J Taylor. Expansion segmentation for visual collision detection and estimation. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 875–882. IEEE, 2009.

[5] Geert De Cubber, Daniela Doroftei, Yvan Baudoin, Daniel Serrano, Keshav Chintamani, Rui Sabino, and Stephane Ourevitch. Icarus: Providing unmanned search and rescue tools. 2012.

[6] Weidong Ding, Jinling Wang, Songlai Han, Ali Almagbile, Matthew A Garratt, Andrew Lambert, and Jack Jianguo Wang. Adding optical flow into the gps/ins integration for uav navigation. In *Proc. of International Global Navigation Satellite Systems Society Symposium*, pages 1–13, 2009.

[7] Gunnar Farneback. Fast and accurate motion estimation using orientation tensors and parametric motion models. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 1, pages 135–139. IEEE, 2000.

[8] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981.

[9] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. Software available at `http://octomap.github.com`.

[10] Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Proc. IEEE International Symposium of Robotics Research (ISRR)*, 2011.

[11] Sung-Yeol Kim, Eun-Kyung Lee, and Yo-Sung Ho. Generation of roi enhanced depth maps using stereoscopic cameras and a depth camera. *Broadcasting, IEEE Transactions on*, 54(4):732–740, 2008.

[12] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE, 2004.

[13] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[14] David Marr, Tomaso Poggio, Ellen C Hildreth, and W Eric L Grimson. A computational theory of human stereo vision. In *From the Retina to the Neocortex*, pages 263–295. Springer, 1991.

[15] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, J-M Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[16] Paul C Merrell, Dah-Jye Lee, and Randal W Beard. Obstacle avoidance for unmanned air vehicles using optical flow probability distributions. In *Optics East*, pages 13–22. International Society for Optics and Photonics, 2004.

[17] Laurent Muratet, Stephane Doncieux, Yves Briere, and Jean-Arcady Meyer. A contribution to vision-based autonomous helicopter flight in urban environments. *Robotics and Autonomous Systems*, 50(4):195–209, 2005.

[18] Randal C. Nelson and Jhon Aloimonos. Obstacle avoidance using flow field divergence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(10):1102–1106, 1989.

[19] MG Petovello, D Sun, G Lachapelle, and ME Cannon. Performance analysis of an ultra-tightly integrated gps and reduced imu system. In *ION GNSS*, pages 1–8, 2007.

[20] Gian F Poggio and Tomaso Poggio. The analysis of stereopsis. *Annual review of neuroscience*, 7(1):379–412, 1984.

[21] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 2009.

[22] Sebastian Scherer, Sanjiv Singh, Lyle Chamberlain, and Srikanth Saripalli. Flying fast and low among obstacles. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2023–2029. IEEE, 2007.

[23] Kahlouche Souhila and Achour Karim. Optical flow based robot obstacle avoidance. *International Journal of Advanced Robotic Systems*, 4(1):13–16, 2007.

[24] Sebastian Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2):111–127, 2003.

[25] Michael Waschbüsch, Stephan Würmlin, Daniel Cotting, and Markus Gross. Point-sampled 3d video of real-world scenes. *Signal Processing: Image Communication*, 22(2):203–216, 2007.

[26] Dong-Wan Yoo, Dae-Yeon Won, and Min-Jea Tahk. Optical flow based collision avoidance of multi-rotor uavs in urban environments. *International Journal of Aeronautical and Space Sciences*, 12:252–259, 2011.

[27] Huili Yu, Randy Beard, and Jeffrey Byrne. Vision-based navigation frame mapping and planning for collision avoidance for miniature air vehicles. *Control Engineering Practice*, 18(7):824–836, 2010.

[28] Simon Zingg, Davide Scaramuzza, Stephan Weiss, and Roland Siegwart. Mav navigation through indoor corridors using optical flow. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3361–3368. IEEE, 2010.

http://www.imavs.org/

Visual Quadrotor Swarm for IMAV 2013 Indoor Competition

Jose Luis Sanchez-Lopez[1], Jesus Pestana[1], Paloma de la Puente[1] and Pascual Campoy[1]

Computer Vision Group
C.A.R. CSIC-Universidad Politecnica de Madrid
Calle Jose Gutierrez Abascal, 2
28006 - Madrid (Spain)
jl.sanchez@upm.es

http://www.imavs.org/

Abstract

This paper presents a whole swarm system for low-cost visual navigation, developed to participate in the 2013 IMAV Indoor Flight Competition. The quadrotor platform employed is the Parrot ARDrone 2.0. ArUco Codes [2] are used to sense and map the obstacles and to improve the pose estimation based on the IMU data and optical flow by means of an Extended Kalman Filter localization and mapping method. A free-collision trajectory for each drone is generated by using a combination of well-known trajectory planning algorithms: probabilistic road maps, the potential field map algorithm and the A-Star algorithm. The control loop of each drone of the swarm is closed by a robust mid-level controller. A very modular design for integration within the Robot Operating System (ROS) [13] is proposed.

1 Introduction

The motivation of the presented work is to design a low-cost framework for quadrotor swarm prototyping. The framework is designed to ease the main issues related to working with a multirobot system, which are obstacle avoidance and partner detection. The focus is to allow the team of designers to try various swarming behaviours on a real robotic swarm, so that the advantages of the proposed strategy can be experimentally demonstrated on an early stage of the development process. In order to obtain a cost-effective testbed the AR Drone 2.0 quadrotor was selected as the aerial swarm agent, as it is a low-cost but competitive platform. The second motivation is to participate in international robotics competitions using simple, but swarming, visual aerial robotic agents. More specifically, the presented work has been implemented to participate on the 2013 edition of the International Micro Air Vehicle (IMAV) Flight Competition.

There exists a large variety of applications which require a robotic system to densely navigate a wide area. Such applications can benefit from a swarming approach for the required data gathering of the problem at task, taking benefit from the robotic population. For instance, such an approach could be applied to security and surveillance of middle sized areas.

The IMAV Flight Competition is one of the most relevant competitions in Europe in the field of Autonomous Aerial Robotics and in the field of Small Remotely Piloted Air Systems (sRPASs). The Computer Vision Group (CVG) participated last year [20], in the 2012 edition, showing the potential and the research experience of the group. The learning experience obtained from the indoor dynamics competition encouraged us to keep working in the same line and also to try a swarming approach on the

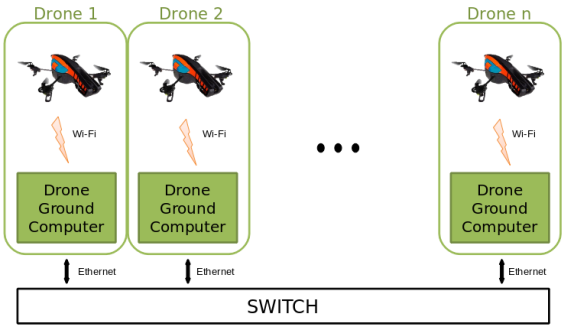


Figure 1: The swarm is composed by identical robotic agents, which are composed of an AR Drone 2.0 which is commanded via Wifi from a ground station. The ground stations can communicate with each other under ROS via LAN, where one of them is running the roscore.

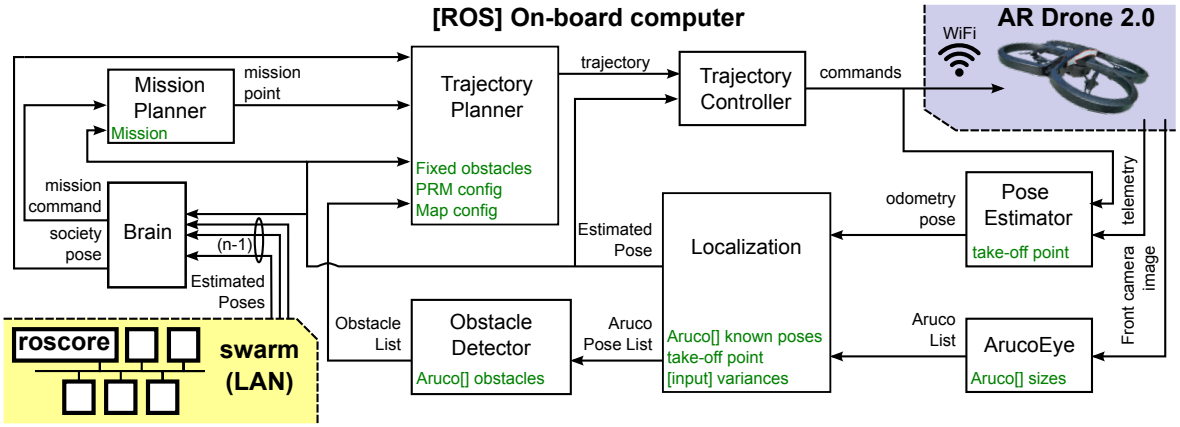


Figure 2: Robotic agent software architecture. The ROS nodes of each agent are executed on a ground station which commands one AR Drone via WiFi. Each white box represents a module, and the green text inside it are configuration files. The localization module is implemented using an EKF which fuses the odometry based estimation with the ArUco visual feedback. This modules broadcasts the estimated pose to the mission and trajectory planning modules, to the controller modules and to the other robotic agents. The brain module receives the estimated position of the other robots and communicates it to the trajectory planner.

2013 edition. Our motivation for participating in such competitions is to develop autonomous systems which can be later modified to perform civilian applications. This year's rules are significantly different respect to 2012 edition. In the IMAV 2013 edition there is only one indoor competition (see [3]) which requires a high level of autonomy. The scenario has some fixed and previously known obstacles (a wall and four fixed poles) and several obstacles located on unknown positions (two windows and four obstacle poles). The indoor competition includes various challenges, among others: flying through a window, flying through the obstacle zone, target detection and recognition, path following and precision landing.

http://www.imavs.org/

After a deep analysis of the contest characteristics, a Visual Quadrotor Swarm was selected as the best option to join the 2013 IMAV indoor flight competition. A swarm composed by a significant number (more than 5) of relatively simple quadrotors (see section 2) is going to be used to achieve all missions except for the dropping one (mission 6). Additionally, as we decided to work with a visual swarm, a external visual aid described in section 3.2 is needed to solve the localization problem. Our swarm is going to be fully automated, and thus the level of autonomy is going to be "Autonomous Mission Control", requiring only one operator to start up and supervise/monitor the whole system (just in case something becomes wrong, the operator could stop independent elements of the swarm, or the whole system).

2 System Description

The system is composed by a swarm of autonomous Unmanned Aerial Vehicles (UAVs). Each drone is autonomous enough to complete a previously defined mission avoiding obstacles and collisions with the other drones of the swarm. There is no high-level intelligence that controls or synchronizes the drones, the system then being a pure swarm. All the drones of the swarm share its pose in order to avoid the partner detection problem.

Each robotic agent is composed of the quadrotor platform and a ground station, see Fig. 1. The agent is composed of separate modules which work using the Robotics Operating System (ROS), see [13]. The Parrot AR Drone 2 [4] was selected as robotic platform, the characteristics of this platform are thoroughly explained in [5]. The ground computer communicates with the drone via Wi-Fi using the ardrone_autonomy ROS package [1]. On the other side, all the drone's ground computers are communicated via ethernet.

3 Drone's Modules

In this section the main modules, see figure 2, of each robotic agent of the swarm are described. The Pose Estimator and the Trajectory Controller modules are explained on the following bibliography [20, 19].

3.1 Drone's Aruco Eye

The localization of the drone in the map is firstly estimated using the Pose Estimatmor module. Since this measure has drift, we need to correct it with an absolute measure when available using the module described in section 3.2. For this purpose, we use a visual external aid: ArUco Codes (figure 3, see [2]). This library allows to calculate the 3D pose of the camera with respect to each ArUco code.

There are two kind of ArUco codes defined in our system. The ones whose position is previously known and the ones whose position is unknown. Some of these ArUco codes are attached to all the obstacles (fixed known obstacles and fixed unknown obstacles).

With this distribution of codes, we are able to locate the drone with respect to the known ArUco codes, and to sense and locate the unknown obstacles with respect to the drone camera.

This visual aid helps us to solve two problems at the same time quite straight forward: we solved the problem of sensing all the obstacles, that is a very hard task using only computer vision, and we also avoid the visual localization problem in a general environment that is not solved precisely yet (even there are some approaches like [11]).

http://www.imavs.org/

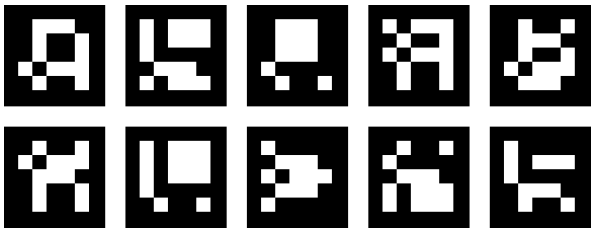


Figure 3: Samples of ArUco Codes, which are markers used to simplify the obstacle detection and localization problems.

3.2 Drone's Localization and Mapping

Localization in indoor environments is a challenging task for UAVs, especially if a low cost and very lightweight solution is required [17, 21, 14, 12]. In the absence of GPS and laser sensors, visual approaches are very popular [21, 14, 12].

In the proposed system, the global localization of each drone is based on the IMU data and optical flow for the pose estimation, calculated by the Pose Estimator module. However, this measure has a drift which may be significant, so it should be corrected with more reliable information from the environment when available. For this purpose, we decided to use visual external aids, the ArUco codes [2] previously described. This library provides the 3D pose of the camera with respect to each ArUco code in a simple and convenient manner. The input of the localization node are hence the pose estimation result (similar to odometry) and the relative observations of the ArUco Codes, received by means of ROS messages.

Since the environment can be partially known a priori, some fixed landmarks are employed. ArUco codes with a priori known global poses are attached to the known poles. Other ArUco codes are placed on the wall and the unknown obstacles, and others are distributed over the floor, with an inclination angle of 45° and mounted on top of a platform 15 cm high. The latter are used so as to improve the visibility of ArUco codes when navigating among the poles; several tests were performed to qualitatively select an appropiate inclination working properly for the ArDrone's camera field of view. Simple and easy-to-use accessories for a fast arrangement of the ArUco codes in the environment were designed and built.

Localization with visual external aids for UAVs has been recently proposed in other works [21, 14, 12]. The method presented by Jayatilleke and Zhang [14] requires all the landmark poses to be known a priori and only works in limited areas, employing quite a simple approach without filtering of any kind. The work by Faig *et al.* presents an interesting approach for local relative localization in swarms of micro UAVs, with the external marks always within the field of view. Our method was mainly inspired by the work by Rudol [21], but our models and formulation are quite different from those proposed by Conte [6].

We designed and implemented an Extended Kalman Filter (EKF) that allows the complete 6 DOF pose of the drone to be corrected by integrating the odometry data and the information from the external visual aids detection. The localization method benefits from the existence of known landmarks, but it also incorporates unknown detected features, using a Maximum Incremental Probability approach for building a map of 6 DOF poses corresponding to ArUco codes positioned in the environment. Similar methods for ground mobile robots were developed in previous work by de la Puente *et al.*, initially based on the observation of 2D point features with a laser scanner [8] and later based on the extraction of planar features from 3D point clouds generated by a tilting laser scanner

http://www.imavs.org/

[10, 9].

In this work, the data association problem does not have to be addressed, since the ArUco readings provide unique ids for the observations and the landmarks. This way, loop closure is facilitated and enhanced robustness can be achieved with a not very cumbersome algorithm which showed nice empirical results in our initial tests.

Non linear state and observation models are used. At each iteration k, the prediction of the pose state x (6 DOF) is given by:

$$\tilde{\mathbf{x}}_\mathbf{k} = f(\mathbf{x}, \mathbf{u})_{\hat{\mathbf{x}}_{\mathbf{k-1}}, \mathbf{u}_\mathbf{k}} = \hat{\mathbf{x}}_{\mathbf{k-1}} \oplus \mathbf{u}, \tag{1}$$

where the $\oplus$ operator corresponds to the composition of relative transformations in the 6D space. The noise in the odometry measurements is considered as gaussian white noise (as required to apply the EKF), and the odometry increment $\mathbf{u}$ is represented as $\mathbf{u} \sim N(\hat{\mathbf{u}}, Q)$.

The observation model is defined by the following innovation vector for an association of observation $\mathbf{o_i}$ and map landmark $\mathbf{l_j}$:

$$\mathbf{h_{i,i+5}} = \tilde{\mathbf{x}} \oplus \mathbf{o_i} - \mathbf{l_j} \tag{2}$$

The correction of the pose state is obtained by the update equation:

$$\hat{\mathbf{x}}_\mathbf{k} = \tilde{\mathbf{x}}_\mathbf{k} - W\mathbf{h}_\mathbf{k} \tag{3}$$

where W is the Kalman gain matrix of the system. The covariance matrices are updated at each stage of the filter as required [22].

The environment is assumed to be static except for the presence of other drones. The accumulation of drift error if the drone is not able to detect ArUco codes all the time may require the incorporation of a forgetting mechanism so that the drone can navigate safely with local maps. In our tests thus far this has not been necessary due to the addition of extra ArUco codes over the floor, but this should be further investigated.

The input parameters of the algorithm (initial pose, covariance values, global poses and ids of the known landmarks) are read from an XML file, by means of the pugixml library [15]. The output is the corrected absolute pose of the drone and the list of global poses of the landmarks belonging to the map. Other nodes subscribe to these topics, as shown in Fig. 2.

3.3 Drone's obstacle generator

Once the position of the unknown ArUco landmarks is obtained, they are processed in order to obtain higher level geometrical features in 2D to be used as obstacles by the trajectory planner. The map of obstacles is rebuilt at every iteration.

To do so, some prior information is required. Each of the obstacles is given a unique id and the ids of the ArUco codes belonging to it are provided. The radius of the poles is known. The poles are modeled with circles given by the coordinates of their center and the radius $c(x_c, y_c, r)$, while the walls are modeled with rectangles given by the coordinates of the center, the width and the length $R(x_c, y_c, w, l)$.

Given the observation of a landmark $\mathbf{l_j}$ belonging to pole i, an initial estimate of the circle i is very easily obtained:

$$(x_{c_i}, y_{c_i}) = \mathbf{l_j} + r\mathbf{dir} \tag{4}$$

with $\mathbf{dir} = (cos(yaw), sin(yaw))$. This initial estimate is further refined by the mean value of incorporating subsequent landmarks belonging to the same pole.

http://www.imavs.org/

The distribution of the ArUcos of the windows has to be known more precisely. Currently, two different options are supported: the first solution is to place the ArUco codes at the corners of each window (with a predefined order) and the second solution is to place one ArUco code at each side of each window and two ArUco codes below each window (also with a predefined order). The second option seems to work best due to the fact that the ArDrone presents a horizontal field of view wider than the vertical field of view. Basic geometry is applied in order to obtain the rectangle models of the wall.

3.4 Drone's Trajectory Planner and Collision Avoidance

The objective of this module is the creation of a free collision 2D trajectory (horizontal coordinates x and y) to achieve a mission.

The module works as follows: a free-obstacles Probabilistic Road Map (PRM) [7] of the 2D map is generated off-line. The advantage of use a PRM instead of using a fixed-cell decomposition is that you can select the number of nodes in the graph and the neighbourhood of them. Also, if the robot is moving thought a zone with a lot of obstacles, new nodes can be added.

Once the free-obstacle graph is created, then an A-Star algorithm [18] searches the path using a potential field map function as the cost of the algorithm. This potential field map is built as a sum of a component that attracts the drone to the end of the obstacle zone and another component that repels the drone from any obstacle. The usage of a search algorithm (A-Star) instead of the potential field map alone [16], avoids the problem of the local minima.

There are three kind of obstacles considered. The first type of obstacles are the fixed and previously known obstacles which are set during the start of the module and are obstacles that never change its previously known position. The second type are the fixed and unknown obstacles that are received from the module described in 3.3 whose position could be different in the time, depending on how precisely are the ArUcos pose determined by the module 3.2. The last type are the unknown and moving obstacles that are other drones and are only considered in the path planning if they are near to the drone. The positions of other drones are received through the brain (see section 3.6).

Once the path is calculated using the A-Star algorithm, it is post-processed in order to obtain a shorter and more direct path, avoiding the noise produced by moving the robot from node to node of the PRM. The post-processing is done using the value of the potential field map. In figure

When new a new pose of the drone or new positions of the obstacles are received, the planner checks if the new obstacles are outside the planned trajectory and if the drone is following the path. Otherwise, the trajectory is re-planned.

With this algorithm we solve the problem of the path planning and the collision avoidance, being able to navigate safely in the map using the Trajectory Controller module.

3.5 Drone's Mission Planner

The mission planner allows the operator to define a mission made for sub-missions and those formed by tasks. The mission definition requires a xml file where the mission is described. It has available different tasks like take off, land, hover, sleep or move.

This module interacts with the trajectory planner (section 3.4) and with the localizer (section 3.2) when moving or with the brain (section 3.6) otherwise.

http://www.imavs.org/

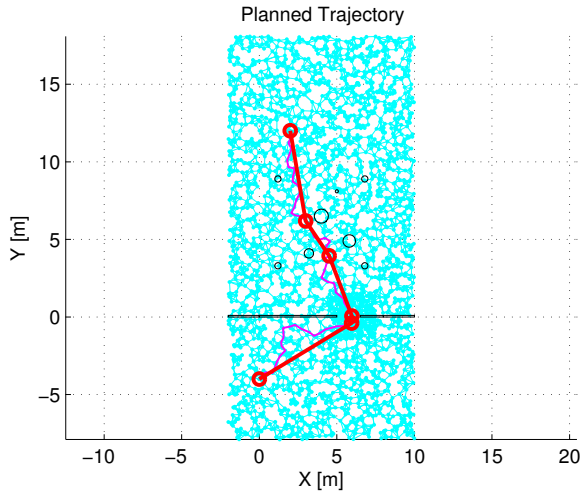


Figure 4: Planned trajectory. In black, obstacles; in blue, the PRM; in magenta, the solution of the A-Star; in red, the post-processed trajectory

3.6 Drone's Brain

This is the very high-level module of each drone. It sends high-level commands to the drone like take-off, land or hover. It also communicates with other drones, receiving the pose of each other drone of the swarm.

The drone's brain communicates with all the modules, being in charge of monitoring each module state, activating or deactivating them.

4 Conclusions

This paper presented an overview of a whole swarm system designed to autonomously complete the indoor mission of the IMAV 2013 competition. The system is low-cost -employing Parrot ArDrone 2.0 quadrotors without any extra sensors- and deployment and setup are quite easy and straightforward due to the fact that only a limited number of known external ArUco codes is required.

The ArUco codes are used for localization and mapping, improving the pose estimation obtained from IMU data and optical flow by means of an EKF based method. The resulting map of ArUco codes is converted to higher level 2D geometrical obstacles used by a fast trajectory planner combining probabilistic roadmaps, the potential field map algorithm and the A-Star algorithm. All the drones have access to the global position of every other drone in the team. The corresponding obstacles are incorporated to obtain a safe trajectory. A robust mid-level controller employs the target global position given by the trajectory planner and the corrected pose of each drone in order to drive them to their respective goals, defined by a mission planner module. The system design and implementation is based on ROS, which makes code sharing, reutilization and monitoring easier.

This paper presents two additional contributions. First, a low-cost framework for visual quadrotor swarm prototyping was described. The framework allows to separate the complexity of some modules such as the localization and obstacle detection capabilities, from the swarm behaviour development and experimental testing. Second, the framework is used to prototype an aerial visual robotic swarm

http://www.imavs.org/

approach to security and surveillance aplications, which is then used to participate on the IMAV2013 competition.

Some initial tests were carried out in simple environments, showing good performance results. We are currently building a real environment like the one shown in the competition rules in order to conduct more complex and realistic experiments and validations. Future work also includes designing and implementing a good graphical user interface for mission management. If the visibility of ArUco codes is not good in the whole area, the localization method may be modified so that old unknown obstacles get removed from the map. The trajectory planning algorithm may be updated with novel ideas.

References

[1] ardrone autonomy ros stack. `https://github.com/AutonomyLab/ardrone_autonomy/`.

[2] Aruco: a minimal library for augmented reality applications based on opencv. `http://www.uco.es/investiga/grupos/ava/node/26`.

[3] Imav 2013 flight competition rules. `http://www.imav2013.org/index.php/information`.

[4] Parrot ardrone 2.0 web. `http://ardrone2.parrot.com/`.

[5] *The Navigation and Control Technology Inside the AR.Drone Micro UAV*, Milano, Italy, 2011.

[6] G. Conte. *Vision-Based Localization and Guidance for Unmanned Aerial Vehicles*. PhD thesis, Linkopings universitet, 2009.

[7] R. Motwani D. Hsu, J.C. Latombe. Path planning in expansive configuration spaces. In *Proceedings of the IEEE International Conference on Robotics and Automation, pp. 27192726*, 1997.

[8] P. de la Puente, D. Rodriguez-Losada, L. Pedraza, and F. Matia. Robot goes back home despite all the people. In *Proc. 5th. Conference on Informatics in Control, Automation and Robotics ICINCO 2008 Funchal, Portugal*, pages 208–213, 2008.

[9] P. de la Puente, D. Rodriguez-Losada, and A. Valero. 3D Mapping: testing algorithms and discovering new ideas with USARSim. In *USARSim workshop, IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[10] P. de la Puente, D. Rodriguez-Losada, A. Valero, and F. Mata. 3D feature based mapping towards mobile robots enhanced performance in rescue missions. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[11] J. Engel, J. Sturm, and D. Cremers. Camera-based navigation of a low-cost quadrocopter. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[12] J. Faigl, T. Krajník, J. Chudoba, M. Saska, and L. Přeučil. Low-Cost Embedded System for Relative Localization in Robotic Swarms. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2013.

[13] Willow Garage. Ros: Robot operating system. `www.ros.org/`.

[14] L. Jayatilleke and N. Zhang. Landmark-based localization for unmanned aerial vehicles. In *IEEE International Systems Conference (SysCon'13)*, pages 448–451, 2013.

[15] A. Kapoulkine. pugixml. `http://pugixml.org/`.

[16] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic, 1991.

[17] G. Mao, S. Drake, and B. D. O. Anderson. Design of an Extended Kalman Filter for UAV Localization. In *Information, Decision and Control, 2007 (IDC'07)*, pages 224–229, 2007.

[18] B. Raphael P. E. Hart, N. J. Nilsson. A formal basus for the heuristic determination of minimum cost paths. *IEEE Transactions on SYstems Science and Cybernetics*, 4(2):100–107, 1968.

[19] Jesús Pestana. On-board control algorithms for Quadrotors and indoors navigation. Master's thesis, Universidad Politécnica de Madrid, Spain, 2012.

http://www.imavs.org/

[20] Jesus Pestana, Ignacio Mellado-Bataller, Changhong Fu, Jose Luis Sanchez-Lopez, Ivan Fernando Mondragon, and Pascual Campoy. A general purpose configurable navigation controller for micro aerial multirotor vehicles. In *ICUAS 2013*.

[21] P. Rudol. Increasing autonomy of unmanned aircraft systems through the use of imaging sensors. Master's thesis, Linkoping Institute of Technology, 2011.

[22] J. De Schutter, J. De Geeter, T. Lefebvre, and H. Bruyninckx. *Kalman Filters: A Tutorial*, 1999.

http://www.imavs.org/

Swarm Exploration of Unknown Areas on Mars Using SLAM

Johanna Matthaei*, Thomas Krüger*, Stefan Nowak* and Ulf Bestmann*

* Institute of Flight Guidance,
Technische Universität Braunschweig
(j.matthaei, th.krueger, stefan.nowak, u.bestmann)@tu-braunschweig.de

Abstract

This work includes a concept for the exploration of Valles Marineris, a wide canyon system on planet Mars. On account of the rough terrain, consisting of caves and mountains, a swarm of different robots is advantageous. In this case two types of robots are utilized: Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs). Due to the distance from Earth to Mars a remotely controlled operation of UAVs is not possible. The target of the project is the development of a robust fault-tolerant system of navigation and the realization of automatic operations. For the environmental detection and mapping laser scanners and a Simultaneous Localization and Mapping (SLAM) concept is used.

http://www.imavs.org/

1 Introduction

Several exploration of the Mars were realised with manually controlled rovers in accessible environment. But especially areas with a rough surface like deep canyons are interesting for the search of water and of extra-terrestrial life. One possibility to explore this terrain is the dangerous and expensive investigation by a manned crew. From the view of technologies perspective this is not yet possible. Another solution are automatic acting robots with a fault-tolerant system of navigation. In this concept two types of unmanned vehicles are used. UAVs and UGVs explore the unknown area as cooperative acting swarm, as depicted in Figure 1.

The Martian scenario has the disadvantage of a dead time between 6 and 60 minutes for radio transmission between Earth and Mars. Therefore a manual control is only possible for slow moving vehicles. So automatic acting is necessary for an adequate speed of exploration and for the control of flying vehicles. A safe exploration of unknown areas also requires an environmental detection and robust navigation, besides routines for automatic operations. Therefore a rover needs information about its states, surrounding terrain and the relative position to other swarm members. A precise navigation is practicable by the combination of self-positioning and relative positioning systems. At this point a comprehensive Simultaneous Localisation and Mapping (SLAM) algorithm is suitable to realise a relative positioning.

1.1 Martian Environment

Especially for UAVs the physical and environmental properties of Mars have to be considered during development. Some relevant parameters are the rotational velocity of Mars with $\omega_{Mars} = 14.62\ \frac{^\circ}{s}$ and the gravity with $g_{0,Mars} = 3.71\ \frac{m}{s^2}$. The Inertial Navigation System (INS) has to be adapted to

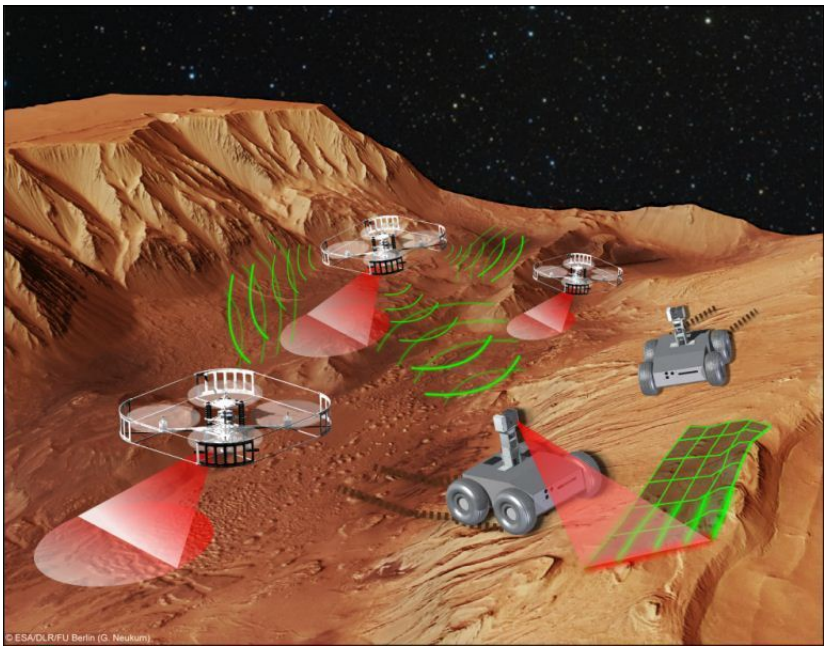


Figure 1: Martian swarm exploration with two types of vehicles

http://www.imavs.org/

these parameters [1].

The atmospheric pressure of the Mars equals the pressure at a terristrial altitude of about 35 km. Depending on the season the variation of the Martian pressure takes place between 400 and 870 Pa[8]. Like in [2] the barometric pressure can be calculated with the following equation:

$$p(h) = 699 \cdot e^{-0.00009 \cdot h} Pa$$

This equation of the atmospheric model is valid for an altitude up to 7000 m. The consideration of the range of a pressure sensor is important. The Mars requires a scale of 10 Pa instead of 1100 hPa like on Earth. Navigation systems on Earth often utilize the global magnetic field. This method cannot be used on Mars because of weak local magnetic fields in scale of tenth of nT [4].

2 Construction of the Vehicles

The two different vehicles are equipped in the same manner. The functional structure consists of a Sensor Readout Unit, a Main PC Unit and a Remote Control Unit which has a conntection to the Actuating Elements, like in Figure 2.

2.1 UAV

The core component of the UAV is an AscTec Pelican Quadrocopter with an actual take-off weight of 1.9 kg. The UAV is equipped with an AscTec Mastermind Board as Main PC Unit and an AscTec AutoPilot, used as Remote Control Unit. Due to the modular construction of the body of the

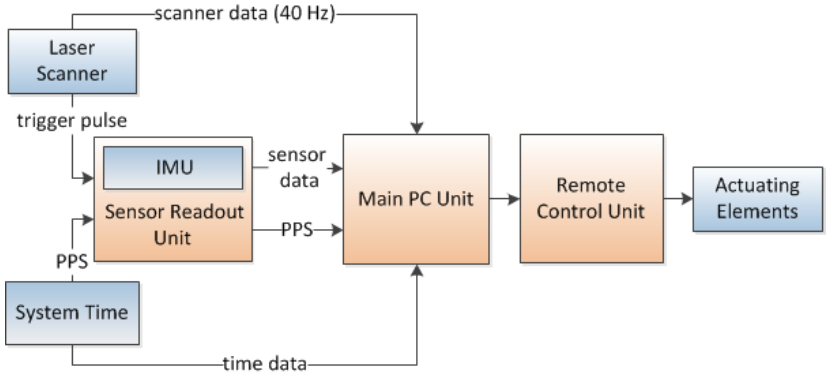


Figure 2: Simplified structure of the UGV and UAV components



Figure 3: Quadrocopter with additional components: laser scanner (red circle), Sensor Readout Unit (green circle)

quadrocopter, additional modules like Sensor Readout Unit and a laser scanner can be mounted. In Figure 3 these additional components are depicted.

2.2 UGV

The UGV constists of a Dr-Robot Jaguar 4x4 with a weight of 20.5 kg and a maximum additional paylod of 30 kg. The chassis is equipped with four wheels, each driven by its own eletrical motor. Therefore skid steering is possible. The body of the rover is designed for outdoor activities and is suitable as experimental vehicle for this project. Most of the electronic components were replaced and new ones were contructed, including the two self-developped boards, named Sensor Readout Unit and Remote Control Unit.

http://www.imavs.org/

2.3 Sensor Readout Unit

The Sensor Readout Unit is based on a STM32F4 microcontroller. It is used for time synchronization and carries the IMU. The time stamping and synchronization is very important for matching data of different sensors. The microcontroller has a clock generating timestamps for incoming signals, among other the Inertial Measurment Unit (IMU) data and the trigger of the laser scanner. The Sensor Readout Unit is also able to trigger e.g. a camera, the Main PC or other sensors. For test scenarios on Earth a GPS receiver is connected with this unit. So a Pulse Per Second-signal (PPS) from the GPS receiver can be used for synchronization. Another task of the Sensor Readout Unit is the transfer of collected data to the Main PC Unit, optional using UART or USB.

2.4 Main PC Unit

The two types of vehicle have a different kind of hardware, but the functionality of this Main PC Unit is the same. As input there are the collected data of the Sensor Readout Unit, the laser scanner and if necessary the information of the GPS receiver. Using these information the Main PC Unit calculates control output values. These are forwarded to the Remote Control Unit which drives the motors.

http://www.imavs.org/

2.5 Sensor Concept

The most important sensors for this concept are the IMU and the laser scanner. The IMU is based on Micro-Electro-Mechanical Systems (MEMS) with high dynamic body-fixed rates and accelerations. In this project the IMU ADIS 16488 from Analog Devices is used. Common sensor errors, like colored noise and temperature dependent drift arise. Therefore a combination with supporting navigation data is essential.

To realize a supporting system all members of the swarm are equipped with a laser scanner. In this case a Hukoyo UTM-30LX(EW) is installed. It is a single-layer laser scanner, which is very small and light weighted. For this reason it is suitable for the integration in UAVs. The technical data of the laser scanner include a maximum range of 30 m, an angular resolution of 0.25°, a detection angle of 270° and an update rate of 40 Hz. With this sensor an environmental detection is only possible in one plane.

Depending on the kind of rover there are two different concepts to obtain a 3D scan. The UGV is equipped with a rotating laser scanner turret. Like in [6] a turret was contructed which is continuously rotating. The laser turret of the UGV moves around a vertical axis. With a rotary encoder the rotation angle is measured and together with IMU data a three-dimensional scan is possible.

On UAVs the laser scanner is mounted vertically, thus a three dimensional scan of the Martian surface is possible but requires a movement of the sensor (see Figure 4). Due to the movement and yawing of the vehicle scan-patterns vary. This change of UAVs position is measured by the IMU, so laser measurements of Martian surface from different perpectives in time and position can be merged.

3 Software Concept

The software concept includes two combined algorithms: the navigation system and a SLAM.

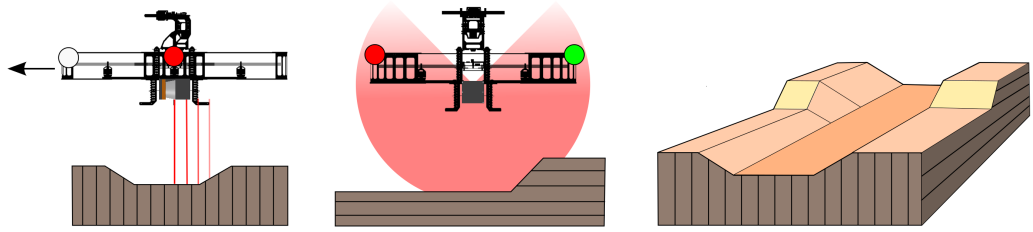


Figure 4: Vertical mounted laser scanner of UAV

3.1 Navigation System

Common navigation systems are using a Global Navigation Satellite System (GNSS) to support the INS, which is based on a IMU. The IMU provides rotational rates and accelerations, but also accumulates errors like colored noise and temperature dependent drift. Since there is no GNSS available on Mars, optical and relative positioning systems have to support the INS to stabilize the positioning. A loosely coupled sensor data fusion of all navigation subsystems is realized. With an Extended Kalman Filter (EKF) corrections for the INS are calculated.

3.2 SLAM Structure

There are a lot of SLAM algorithms in the field of robotic, but established algorithms offer a base for relative positioning to nearby obstacles. Usually the laser scanner is mounted horizontally and has consequently an ideal perspective for indoor scenarios and structured environment. Landscapes, like the surface of the Mars, feature only a small quantity of distinctive objects, which can be scanned horizontally. Thus the sensor is mounted vertically on the UAV and the data processing needs a translation or rotation of the vehicle to get the contour of the surface, as described in 2.5.

In [3] a concept of combining two data sources to one three-dimensional map is mentioned. An orbital a priori map is matched with a rover based scan. The main focus in that work is on long-range localization of the rover for distances about 10 km and more. The idea of a vertical mounted laser scanner on the UAV is used among others in [7]. The method was used to generate a terrain model, but in this case GPS signals are available in scenarios on Earth.

The idea in this work includes three components of data sources: A rough a priori map of Martian surface, environmental data of UGVs from the ground and data of UAVs from above. Surface information in scales of meters are already available, based on recordings of Mars orbiters, and allow an approximate orientation. Each swarm member collects sensor data, including IMU data and time stamped laser scanner data independently. Two different perspectives, which complement one another are generated. The target is to refine the rough Martian surface information with new measurement data detected by the different rovers. Using a communication link between all robots distinctive landmarks can be matched and a collective map is shared.

One requirement of mapping the Martian surface, is a volumetric representation of the space. There are already several methods and libraries of source code. One possibility is the use of OctoMap [5], an effective method of data processing with probabilistic occupancy estimation. The base of this

http://www.imavs.org/



Figure 5: The test vehicles

processing are octrees, a hierachical data structure. Each node of the structure represents a cubic volume, which can be recursively subdivided into eight subcubes. Therefore the same map can be recalled in different levels of resolution.

4 Results

After finishing the construction of the test vehicles some test flights have been done (see Figure 5). The scenario was on a local model plane airfield and as test object a van was chosen (Figure 6 left). The goal of the first tests is the evaluation of the ability to merge laser measurements of both vehicle types.

With the UAV the van is scanned from above. A straight flight over the car with an altitude of approximately 5 m is executed, while the ground vehicle was measuring from a fixed point of view nearby the van. The result is shown in the middle of Figure 6. The blue point cloud represents the meausurment of the UAV whereas the red point cloud is obtained from the ground vehicle. The combination of the two scans generates a clear contour of the scanned van. The combined point cloud of UAV and UGV is pictured on the right of Figure 6 visualized with OctoMap.

5 Conclusion

The theoretical exploration of the Mars is the main target of this project. In this work a concept of navigation and mapping in unknown areas is introduced, utilizing a swarm of robots, which are equipped with laser scanners. The two vehicles offer different perspetives of measurement and can complement one another. This kind of environmental detection enables the base for detailed mapping. Due to data fusion of INS and laser scans, drift parameters of the IMU can be estimated and compensated via an Extended Kalman Filter. Therefore a higher precision of positioning is possible.

http://www.imavs.org/

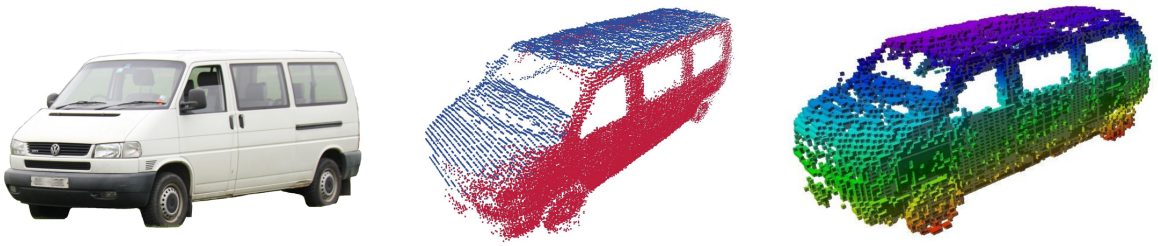


Figure 6: A picture of the scanned Object, the visualization of two point clouds and the tranformation in OctoMap

6 Acknowledgements

This work has been performed within the Valles Marineris Explorer (VaMEx) project, which is partly funded by the Federal Ministry of Economics and Technology administered by DLR Space Agency (FKZ 50NA1212) and the Bavarian Ministry of Economic Affairs, Infrastructure, Transport and Technology administered by IABG GmbH (ZB 20-8-3410.2-12-2012).

References

[1] Raymond E. Arvidson. Mars Gravity Models. *NASA. http://geo.pds.nasa.gov/dataserv/gravity_models.htm 17.05.2013*, 2013.

[2] Tom Benson. Mars Atmosphere Model. *NASA. http://www.grc.nasa.gov/WWW/k-12/airplane/atmosmrm.html 17.05.2013*, 2010.

[3] Patrick Carle and Timothy Barfoot. Global Rover Localization by Matching Lidar and Orbital 3D Maps. 2010.

[4] J. E. P. Connerney, M. H. Acuna, N. F. Ness, G Kletetschka, D. L. Mitchell, R. P. Lin, and H. Reme. Tectonic Implications of Mars Crustal Magnetism. In *Proceedings of the National Academy of Sciences*, volume 102, ISs. 42, pages 14970–14975, 2005.

[5] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[6] Gregor Michalicek. *Development of a 3D SLAM Exploration System.* PhD thesis, Universität Hamburg, Hamburg, 2010.

[7] Sebastian Thrun, Mark Diel, and Dirk Haehnel. Scan Alignment and 3-D Surface Modeling with a Helicopter Platform. *International Conference on Field and Service Robotics*, 2003.

[8] David R. Williams. Mars Fact Sheet. *NASA. http://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html 17.05.2013*, 2010.

http://www.imavs.org/

Technical Session 5

Structure / Flapping wing

Chairman : Hans Goosen

- Guillaume Bavoux, Baptiste Delannoy, Benoit Wach and Renaud Kiefer. ***Development of a Quadrotor based on an Innovative Composite Shape with Carbon and Kevlar Structure***
- S Deng, B.W. van Oudheusden, B. Remes, R. Ruijsink and H Bijl. ***Experimental Study of "Delfly Micro" in Forward Flight***
- Johannes Goosen, Hugo Peters and Qi Wang. ***Resonance Based Flapping Wing Micro Air Vehicle***

http://www.imavs.org/

Development of a Quadrotor based on an Innovative Composite Shape with Carbon and Kevlar Structure

Baptiste Delannoy*, Benoit Wach†, Renaud Kiefer‡, Achraf Haout§, Maxime Collette¶ and Vincent Person‖

INSA Strasbourg, Strasbourg, France
baptiste.delannoy@insa-strasbourg.fr
benoit.wach@insa-strasbourg.fr
renaud.kiefer@insa-strasbourg.fr
achraf.haouet@insa-strasbourg.fr
maxime.collette@insa-strasbourg.fr
vincent.person@insa-strasbourg.fr

http://www.imavs.org/

Abstract

This project is included in Mechatronics engineering education at the National Institute of Applied Sciences (INSA) of Strasbourg. It consists in designing, building and controlling a Micro Aerial Vehicle with quadrotor structure. In this article, the design, manufacturing and control approach is detailed.

The first part deals with the choice of an innovative shape and the research of a reliable material. The structure is composed of two rotational symmetric elements, made of a three-layer sandwich composite material, based on carbon-kevlar tissue and airex, in an epoxy matrix. This composite provides excellent mechanical properties for such applications. Its special shape allows arms work in traction and compression, in conjunction with traditional bending, and demonstrated great strength, a very good vibration absorption and a very weak distortion at the end of the arms. Moreover, the structure embraces and protects the embedded electronics while allowing free access to components and a modularity of the elements that provides easy and fast intervention. The minimalist and uncluttered characteristic of the structure gives us a very good compromise between protection and weight. The two main parts of the structure, strictly identical, were produced by vacuum lamination with a mold realized on a Computer Numerical Control machine. The two elements of the structure are ready for use and provide an easy assembly thanks to the direct integration of fastening systems.

The second part of this article talks about embedded electronics and control. First of all, a Matlab Simulink model has been developed and allows the development of the automatic level control. The control structure is implemented on a microcontroller programmed in C code. The fine adjustment of the servo control is realized in an empirical way, thanks to a first reliable prototype of the quadrotor and a test bench developed by the team.

*Member of the mechanic team
†Member of the mechanic team
‡Supervisor of the project
§Member of the electric team
¶Member of the electronic team
‖Member of the electronic team

http://www.imavs.org/

1 Introduction

This project have been developed in the INSA Strasbourg's school, a french engineering school based in Strasbourg. It began two years ago with the aim of designing and creating an innovative quadrotor. The first team was composed by six members who worked on this project as a part of the mechatronic cursus (four hours per week). A new team of five members took over the project as an internship this summer in order to participate in the IMAV competition. The following pages will presented the work done by both team without distinction.

This paper will be divided such as presented below :

1. The birth of the idea
2. The mechanic part
3. The electronic part

2 The birth of the idea

The main idea was first putted on paper. Then a 3D model was rapidly created :



Figure 1: The initial idea

The next step was to determinate the material and the process of fabrication of such a UAV knowing that this two aspect are bound.

http://www.imavs.org/

3 The mechanic part

3.1 Determination of the structure material

We rapidly agree to use a composite material to create this UAV because this family of material possessed the advantage to often be light and strong. With the CAD, we did some simulation to test the best material :

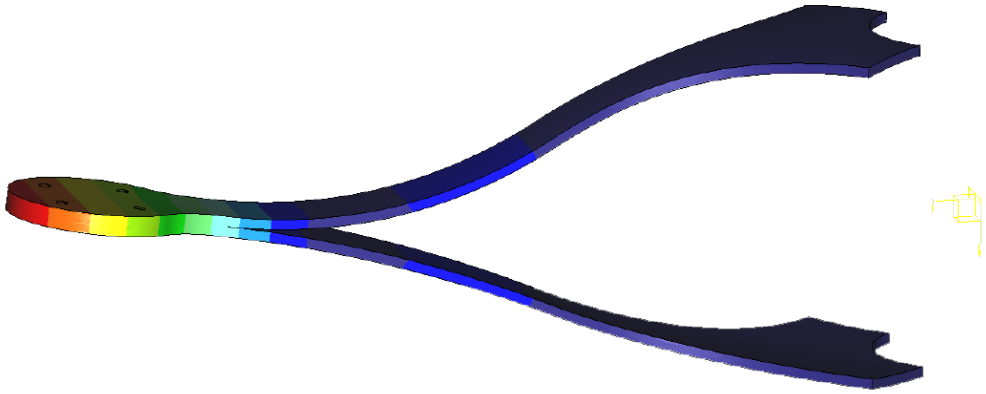


Figure 2: Deformation measurement

The major displacement (here in red) is 1.934 millimeters. So we compared this result to some that previous year had on a classic shape done with an aluminum structure. The major displacement on this previous UAV was about 4 millimeters. The displacement was reduced by 100%. This solidity is due to the fact that the two part help each other to compensate the displacement.
In conclusion, we decided to use a sandwich composite composed by a layer of Airex (a closed-cell foam) between two layers of Carbon-Kevlar.

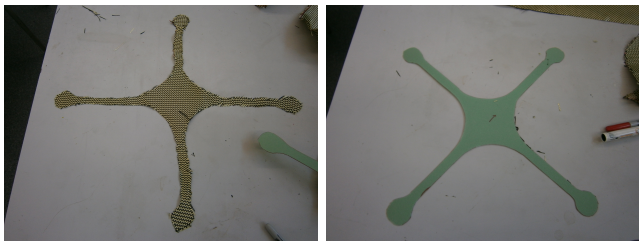


Figure 3: Carbon-Kevlar layer and airex layer

Those tests also helped us to determine the best thickness of the Airex foam and the best width for the arm.

http://www.imavs.org/

3.2 The fabrication process

As the material was determined, we decided to create the structure by using a vacuum lamination process. The mold needed have been modeling in 3D. This 3D model have then been used with a CAM software in order to machine the mold on a CNC Milling Machine. The material of the mold is HDF (High-density fiberboard), an engineered wood product. The mold is then covered with gel-coat and is ready for molding.

The following pictures show the different steps of the fabrication process :

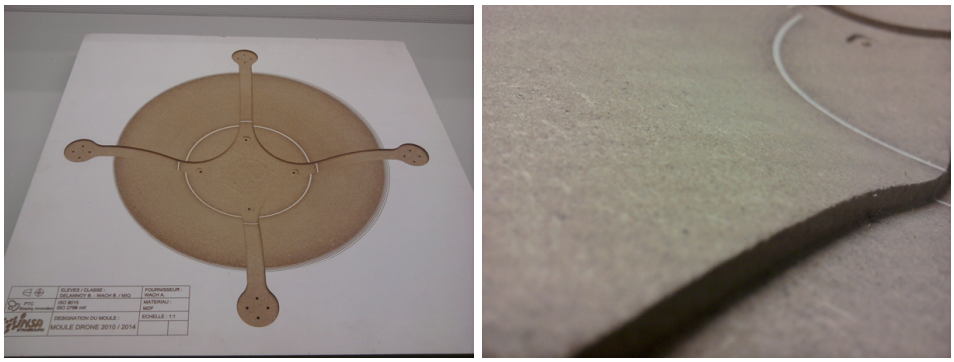


Figure 4: The mold after machining and its surface state

After the application of a release wax, some epoxy resin is applied and the first layer of carbon-Kevlar



Figure 5: The different steps of the fabrication process

is putted on it. Between each layer, the epoxy resin is applied. After the first layer of Carbon-Kevlar, we apply the Airex foam and some Carbon to solidify the weak point. Finally the last layer of Carbon-Kevlar is applied. A microporous membranes and some felt are putted on it to absorb the epoxy excess. Then the vacuum is made to give the shape. After approximately 24 hours, the structure is ready.

3.3 The landing gear

The landing gear was designed the same way of the rest of the structure :

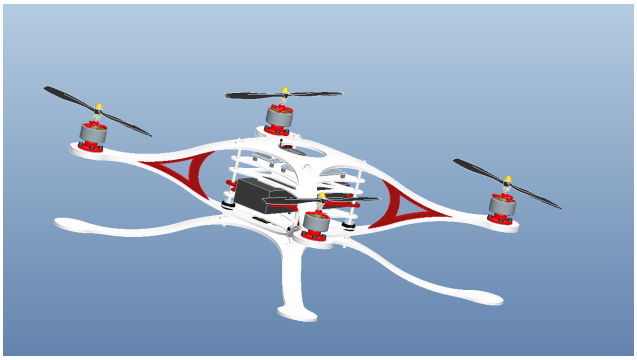


Figure 6: The UAV with its landing gear

But a problematic rapidly appeared after the first flights. The material used to build the structure is to easily deformed and become too rapidly plastic. The problem appear due to the fact that this structure was initially created to be a two-pieces structure with the two pieces supporting each other. We decided to change the material but not the shape, so we could keep the same fabrication process.

We did some tests to determine the best material to create the landing gear. To do so, we made different test specimens :

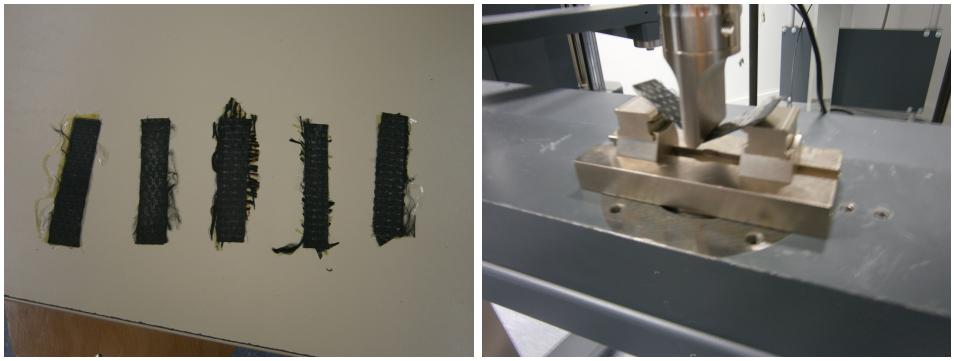


Figure 7: The test specimens and a specimen during the test

This test have been made to determine which material must be used and the number of layers to have a landing gear strong enough to not be deformed after the shock but flexible enough to amortize the shock.

The results obtained led us to the conclusion that we should use Carbon only because the Carbon-Kevlar present less advantage for what we need for the landing gear. We also obtain to use four layer of Carbon for the optimization.

http://www.imavs.org/

3.4 The electronic support

The shape of this UAV also present the advantage to have all the electronic inside the structure in order to protect it while keeping the ability to change it rapidly.
Each board is cut with a CNC Milling Machine and is design for different purpose while keeping the same shape. The first board, called "the command board", supports all the command electronic. The second part, called "the battery boards", are all the boards that support and block every movement of the battery. The last board supports the different components necessary for the competition.

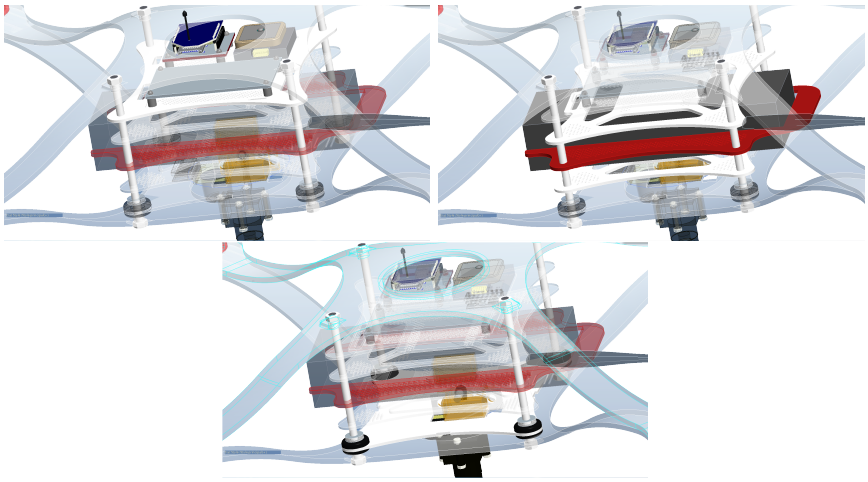


Figure 8: The three different boards inside the UAV

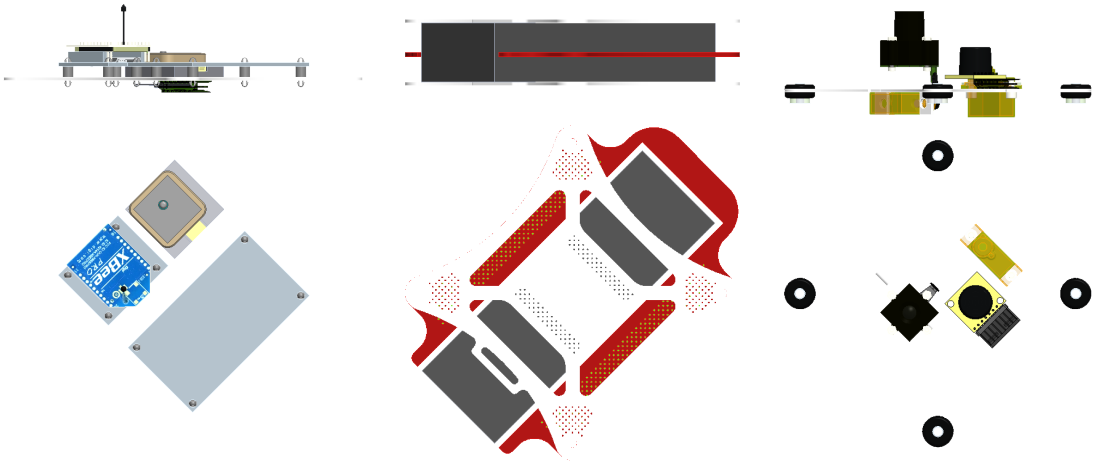


Figure 9: The three different boards

http://www.imavs.org/

http://www.imavs.org/

4 The electronic part

4.1 the electronic architecture

The electronic part has the following structure:

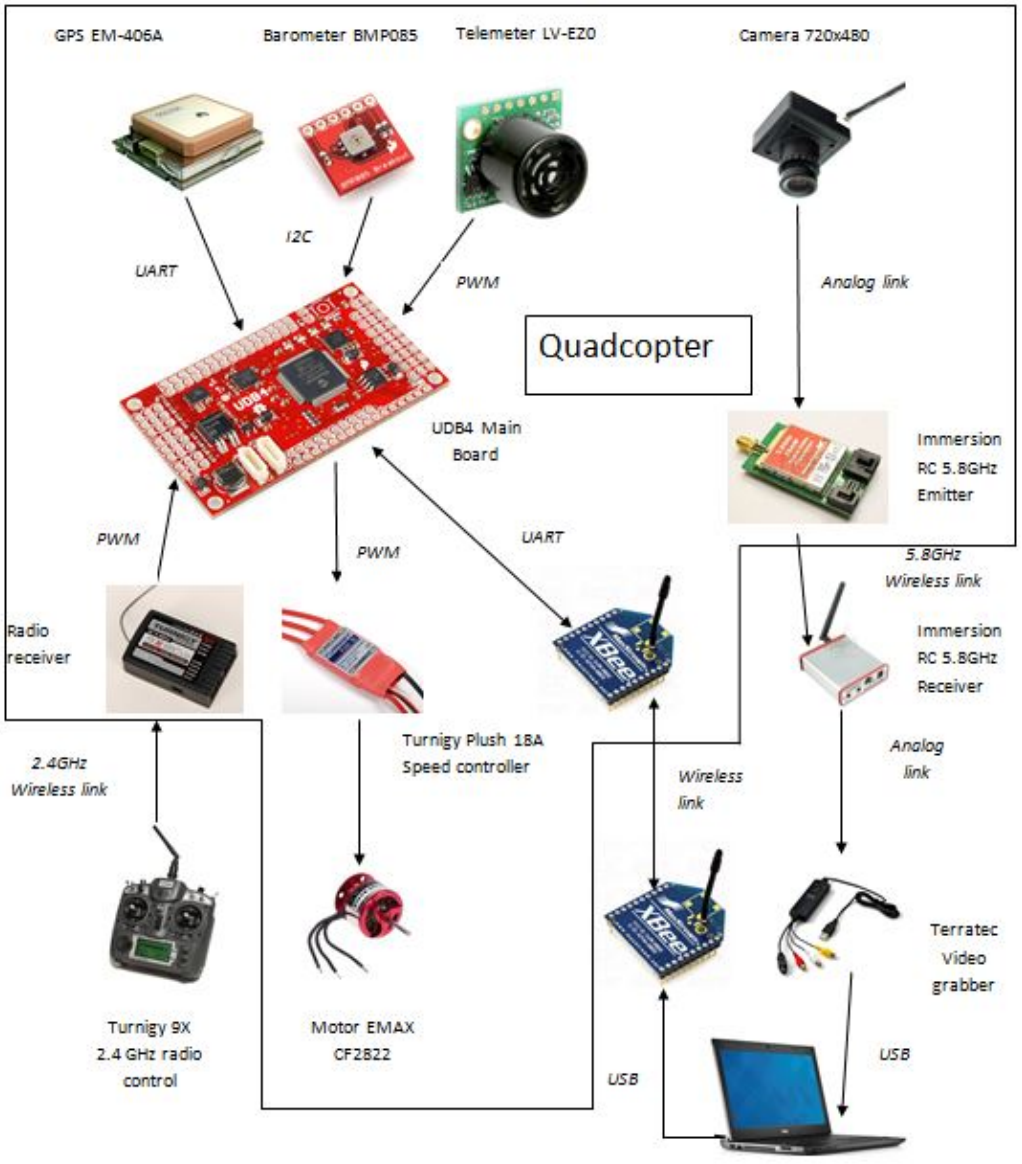


Figure 10: The electronic architecture

The central element is the UDB4-board. It contains accelerometers and gyroscopes to compute the UAVs orientation. The microcontroller used is a dsPIC33FJ256GP710. To this board are connected three other sensors: A GPS module, a barometer and an ultra-sound telemeter. GPS is used for navigation. Barometers measures are used to compute the altitude of the UAV (+/- 50cm). The telemeter is used on take-off and on landing to have a better altitude control. Telemeter is also the altitude sensor for low-altitude flying (under 2 meters). The motors are brushless motors. They need a specific controller which converts DC current from the battery into 3 phases AC current. The controller receives PWM signal from the UDB4. 1ms pulse width means no rotation and 2ms pulse width means maximum speed. The frequency of the current in the motor controls the speed. Control of the UAV is made in 2 ways: First, an 8-channel radio-system allows user-control of throttle, roll, pitch and yaw. An additional channel enables or disables the automatic flying. Second, a XBee link between the UAV and a ground PC is implemented to receive information from the UAV (GPS-Position, altitude, mission state). The embedded camera sends images through a 5.8 GHz video emitter. The ground computer receives and processes images to detect and decode QRCodes for example. The vision part uses the XBee link to send to the UAV information about a target position, so that the UAV places itself on it. The power source is a 3S LiPo battery with 5Ah Capacity. Flying time is between 5 and 10 minutes depending on flying condition like wind.

http://www.imavs.org/

4.2 The program

The program running on the USB4 is the gentlenav program with some modifications. The main part of the program is the stabilization of the UAV. It is made through a PIDD regulator on the position. We added functions to tune the coefficients from a PC through the XBee link. The correct coefficients have been found by tests first on a test platform and then by flying. We added altitude regulation with the ultrasound telemeter. It consists in 2 steps: First, a position error computes a vertical speed. Then, regulation of the vertical speed applies correction on the speed of all the 4 motors. We added also the barometer by using the provided I2C driver. Temperature and pressure measures are used to compute the altitude.

4.3 The vision system

Vision part needs an independent computer on the ground station. Image from the camera are transmetted through a 5.8GHz radio system and sampled through an analog video grabber. For the image processing, we use the open source image processing library OpenCV. The analog grabber is recognized by OpenCV as a webcam. QRCode decoding implements the ZBar library.

5 Conclusion

In this paper, we have tried to synthesize the work done by the two teams presented in the previous pages. The main point of our UAV is the innovative shape and the innovative use of the Carbon-Kevlar in this kind of UAV. After all the tests and all the reasoning, an UAV is now ready to fly and all the results are satisfying.

Thanks :

The team wants to specially thanks the IMAV 2013 edition to allow us to present our work.

The INSA Strasbourg *'s school, for its support in this project and more particulary M. Kiefer, supervisor of the project.*

All those who have contributed to this project.

http://www.imavs.org/

Experimental Investigation of the Flapping Performance on 'Delfly Micro'

S. Deng, B.W.vanOudheusden, B.D.W.Remes, M.Percin, H.Bijl and H.M.Ruijsink

Delft University of Technology, Delft, the Netherlands
S.Deng@tudelft.nl

Abstract

This study addresses the quantification of the aerodynamic performance of the 'Delfly Micro', which is currently the smallest ornithopter in the 'Delfly' family with an X-Wing configuration. Aerodynamic forces were measured on a full-scale 'Delfly Micro' by a sophisticated balance system. In view of the influence of the wing flexibility on the performance, wings with varying membrane thicknesses and stiffener orientations were examined in hovering case. To characterize the forward flight regime, aerodynamic forces were also measured with varying free stream velocities and angles of attack in wind tunnel. The results indicate that wing flexibility indeed influences the aerodynamic performance, where a more rigid wing can produce more force, whereas with a lower force to power ratio. According to the wind tunnel data, the 'Delfly Micro' can attain a flight speed up to 3 m/s, at a flapping frequency over 30 Hz and angle of attact of 30 degrees.

http://www.imavs.org/

1 Introduction

FMAVs (flapping wing Micro Air Vehicles)have gained interest due to their potential in flight efficiency and maneuverability. Since 2005, the MAV group at Delft University of Technology has engaged in the development and research of FMAVs regarding the related technologies in aerodynamics, mechanism, electronics and autonomy, etc. Our group has developed three types of ornithopters different in size:' Delfly I', 'Delfly II' and 'Delfly Micro'. It is the Delfly development philosophy to keep minimizing the size of the Delfly without sacrificing the flight performance. Previous studies[1, 2, 3, 4, 5] during the past years have focused on 'Delfly II. In this paper, we turn the attention to the smaller 'Delfly Micro, which only weighs 3 grams and has 10 cm wing span. With a 30mAh lithium polymer battery, it can fly for 3 minutes with a 50 m range. This small size makes it the smallest flying ornithopter with onboard camera in the world. Furthermore, the 'Delfly Micro' is an ideal platform for studying the aerodynamics of low Reynolds flyers in nature.

In this study, experiments on the aerodynamic performance of the real scale of 'Delfly Micro have been conducted to obtain information on the preliminary flight envelop, including hovering and level flight. Flexibility of the wing significantly effects the overall aerodynamic performance of FMAV, as has been shown for Delfly II[5]. Therefore, three different membrane thicknesses (2, 5 and 10 micrometer) and three different stiffener layouts are examined under hovering flight conditions. A high speed camera is placed in front of the 'Delfly Micro' to detect the real-time phase angle during the flapping motion. The information from these images allow the aerodynamic forces to be synchronized with the corresponding wing position, hence, the phase in the flapping cycle. Further measurements are carried out to characterize the level flight with oncoming free stream. The wind velocity and angle of attack are varied between $2 \sim 4m/s$ and $20° \sim 40°$, respectively. The forces and frequency detector signal are recorded instantaneously by the FPGA system, together with the motor current and voltage, to determine the power input.

http://www.imavs.org/

This paper is organized as follows. In section 2 the experimental setup is presented. Subsequently, the results of the experimental and its analysis are described in section 3, followed by conclusion in section 4.

2 Experimental Setup

2.1 Full-scale 'Delfly Micro'Model and Force Sensor

In order to mimic the real flight performance of the 'Delfly Micro', the model we used is a 1:1 scaled model, as shown in Fig. (1Left)). Clap-and-fling had been proven as one of the high lift mechanisms in flapping wing flyers[6]. In order to generate more force during flight, 'Delfly Micro has a flapping angle of 120° which gives three clap-and-fling occurrences during a flapping cycle, at both sides and at the top. To obtain a more reliable and stable RPM operation during the measurement, the actual DelFly Micro motor is replaced by a brushed DC motor (diameter $\phi = 7$mm) that is actuated by an external power supply. The flapping frequency is captured by an optical resistance mounted on both sides of the main gear.

The force sensor used is an ATI Nano17 Titanium, which is the smallest 6-axis sensor for commercial use and which can resolve down to 0.149 gram force without filtering. A DC motor is coupled to the force balance to give a precisely changeable angle of attack in the range of $-15°$ to $35°$. The measured forces are used to compute the resultant vertical force (Lift= T$sin\alpha$ +N$cos\alpha$) and the resultant horizontal force (Thrust=T$cos\alpha$-N$sin\alpha$) acting on 'Delfly Micro', see Fig (1Right)).

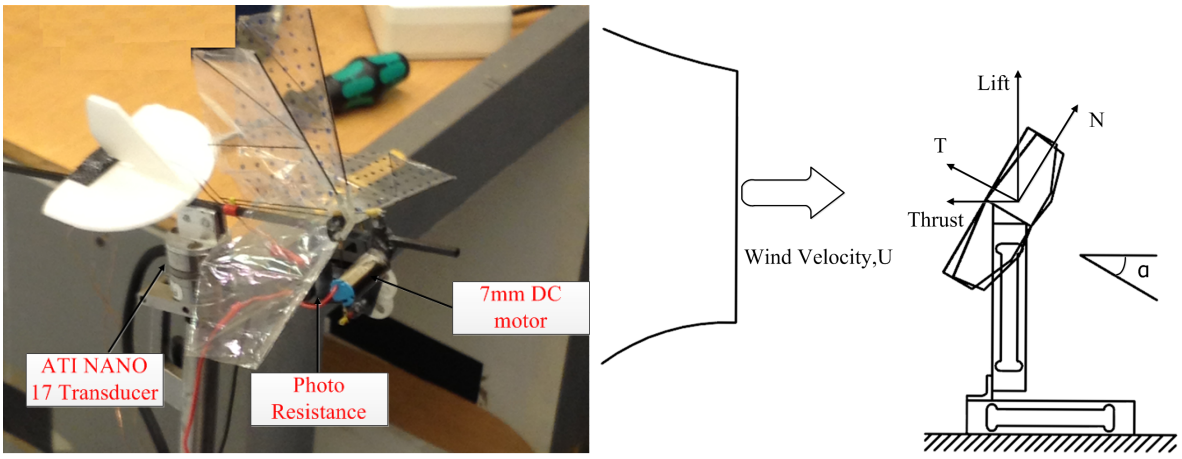


Figure 1: 'Delfly Micro' Model and its force relation on force balance. Left: 'Delfly Micro' Model; Right: Force on force balance.

2.2 Wind Tunnel

The measurements with free stream are conducted in the open jet W-Tunnel at the Delft University of Technology (DUT), shown in Fig.(2). The measurement section is a square section with 0.6m×0.6m, which is large enough to prevent the jet shear layers' interaction with 'Delfly Micro'. The contraction

ratio is 3.62 and a maximum flow speed of around 15 m/s can be obtained[7]. In order to detect the incoming flow velocity, a calibrated hot wire velocity system is placed in the nozzle.

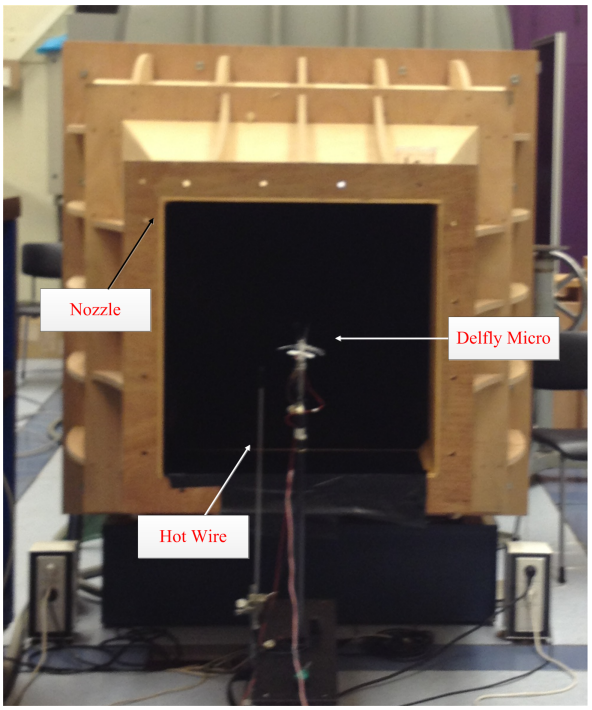


Figure 2: W-Tunnel in DUT

2.3 Wing Construction

Fig. 3 shows the standard wing layout used on 'Delfly Micro', with the wing span of 100 mm and aspect ratio of 3.2. The reason for choosing this configuration as the starting point is that such wing shape and stiffness orientation has been empirically optimized as the most efficient wing for the DelFly II[5].

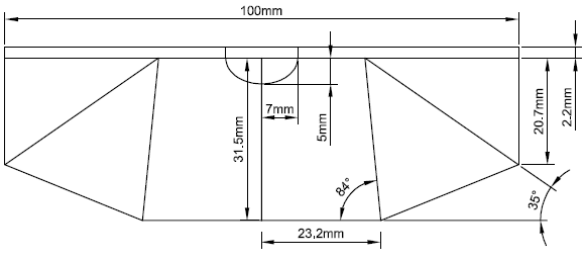


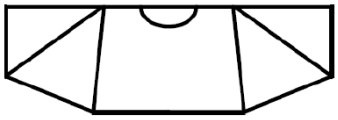
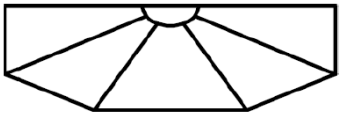
Figure 3: Schematics of 'Delfly Micro' Wing

In order to address the effect of flexibility on the wing, 6 sets of wings with varying membrane

http://www.imavs.org/

thickness and stiffener orientation were built, as summarized in Table.1. The leading edge and stiffeners in all the wings are carbon rods with ϕ=0.5mm and 0.15mm, respectively. Furthermore, the wings are in-house built, following the technique developed by Bruggeman[5], using a vacuum table to guarantee the symmetry and uniform tension distribution on Mylar membrane.

Table. 1 Wing Properties

Name	Wing Layout
2_Clean	
5_Clean	
2_Std	
5_Std	
10_Std	
2_DiffStiff	
Note: the first number in the name of the wings represents the Mylar thickness in μm. **Clean** indicates the wings without stiffeners **Std** is Standard Wing. **DiffStiff** is the wings with different stiffeners orientation.	

3 Results

3.1 Phase Distribution and Force Synchronization

To determine the phase distribution, i.e. the wing position variation over the flapping cycle, a high speed camera is placed in front of the Delfly Micro. 1000 images are taken at an acquisition frequency of 3000Hz. The Canny method is employed to detect the edges, because it is less sensitive to noise. The Hough transform is then applied to detect the leading edges in the binary images. The obtained phase distribution is fitted by a 6 order polynomial curve fit. Leading edge recognitions are performed on the 2_Std Wing for different flapping frequencies, seen in Fig. (**??**). Aerodynamic forces are synchronized via locating the triggered signal for the camera from the optical sensor. Fig. (4(Right)) shows the synchronized forces corresponding to Fig. (4(Left)) for different frequencies. It can be observed that, when increasing the flapping frequency, the thrust production is increased. Furthermore, the peak in generated thrust shifts with the frequency, which had also be observed in the research of Delfly II[7].

http://www.imavs.org/

The explanation of this shift phenomena is as yet not explained and remains an interesting topic for further investigation.

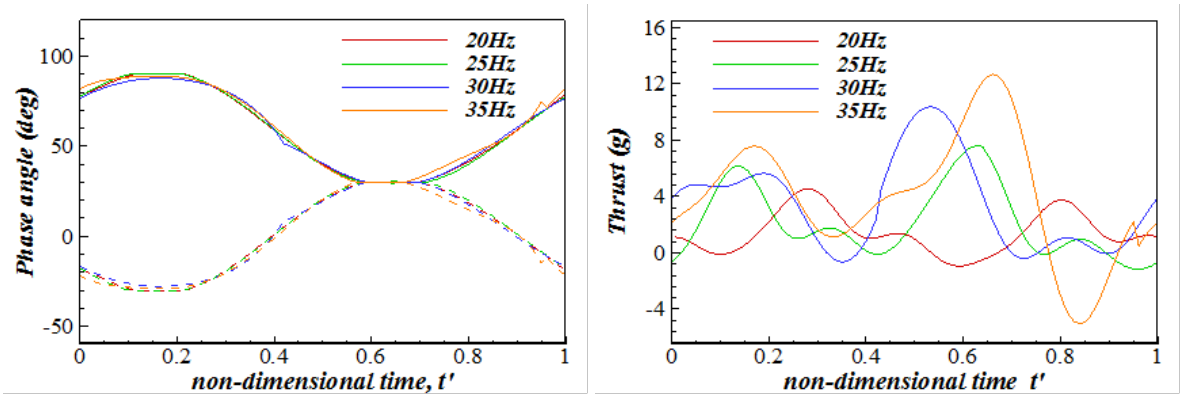


Figure 4: Phase distribution and force history during the flapping cycle. Left: Phase distribution of upper and lower wing, solid line is the leading edge of the upper wing, lower wing phase angle is represented with dashed line; Right: Forces from different flapping frequencies over the flapping cycle.

3.2 Effect of Wing Flexibility

Investigation of the influence of wing flexibility is conducted on the 6 sets of wings in Table.1. Thrust force is converted to grams and the power consumption can be obtained by Joules law $P = I_{measured} \times V_{measured}$. Furthermore, the ratio of thrust production to power consumption (T/P), which is a critical driving parameter in design optimization, is examined to evaluate the performance of the wings.

Tested results are plotted in Fig. (5). It can be seen that the clean wing configurations require the smallest power input when compared with the stiffened wings. However, the low thrust generated by the clean wing results in an inefficient configuration in view of the thrust to power ratio, as shown in Fig. (5(Right)). The wings with stiffeners and 5 and 10m Mylar foils produce almost the same thrust, at a given flapping frequency. Wing 2_Std has a lower thrust production as may be expected from the increased amount of flexibility. Fig. (5(Middle)) illustrates that the more rigid wing required more power input respect to certain frequency. From Fig. (5(Right)), it follows that all the stiffened wings have better thrust to power ratio than the clean wings, and that Wing 2_Std possesses the best performance. Wing 2_DiffStiff displays the original wing stiffener configuration that has been used on the 'Delfly Micro' in the past, and which had also been examined in present test for comparison. One can see that this initial wing generates less thrust than the other stiffened wings. Meanwhile, it is more power consuming and a poor T/P performance of the 2_DiffStiff wing results, as can be seen in Fig. (5(Right)). Overall, within the enveloped flapping frequency of 'Delfly Micro' regions from 25Hz to 35Hz, the standard wing built with a 2 m foil displays the best performance, in terms of the highest thrust to power ratio. Note that the thrust generated around 30Hz is around 3 grams, which is equal to the total weight of 'Delfly Micro'. That means 'Delfly Micro' is potentially able to hover. However, it has been observed that it cannot hover in real flight test, perhaps because the aerodynamic force and the weight distribution always produce a pitch moment in real flight. Further optimization has to be carried out on the current 'Defly Micro' configuration to achieve hover ability.

http://www.imavs.org/

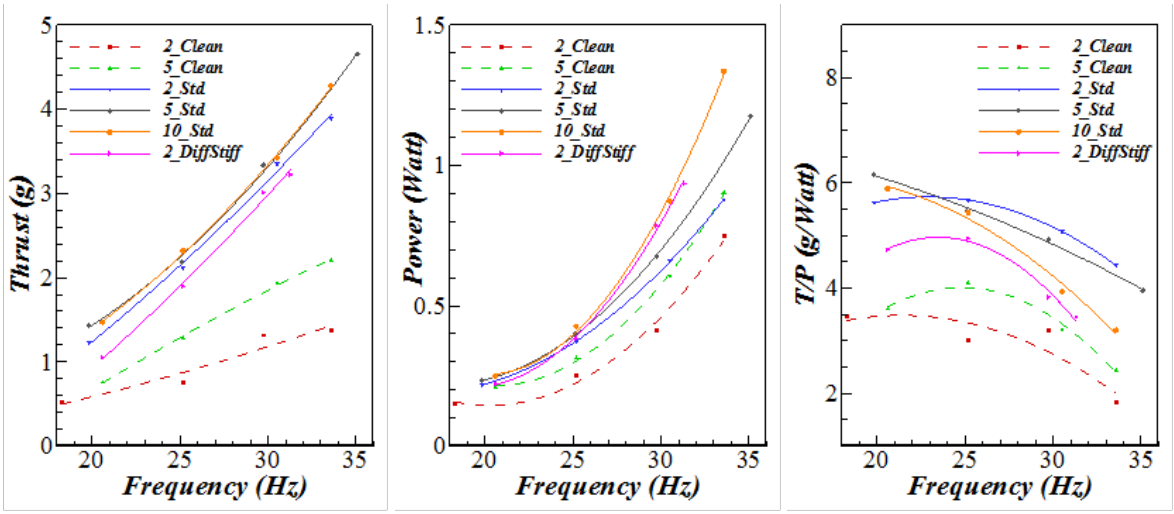


Figure 5: Performance plots for wings with different flexibility. Left: Thrust; Middle: Power consumption; Right: Thrust to power consumption ratio.

3.3 Wind Tunnel Test

The flapping performance for the hovering case has been experimentally investigated as discussed in the previous section. This section will examine the flapping performance within incoming flow velocity, representing the forward flight regime. The calibrated wind speed is set at 2m/s, 3m/s and 4m/s. For each velocity, the angle of attack changes from 20° to 40° with an increment of 10°.

Fig. (6) shows the lift and thrust with varying frequencies and free stream velocities. The horizontal and vertical dash lines represent the weight of 'Delfly Micro' (3.07g) and level flight at a constant velocity (zero thrust), respectively. These two lines segment the figure into four parts, where the right top quarter is the region where 'Delfly Micro' has more lift than its weight and positive thrust for flight. The crossing point of the two lines is the equilibrium level flight point of 'Delfly Micro', where the lift equals the weight and zero thrust is zero (actually, this is the total horizontal force, so thrust minus drag in the conventional concept of aeronautics). For the investigated range of flapping frequency and angle of attack, some points fall within the flight operation region for U = 2 m/s and 3 m/s, but this is not the case for U=4m/s. This suggests that 'Delfly Micro' may not be able to fly faster than 4m/s. Moreover, the level flight points can be found in Fig. (6(Left)) that 'Delfly Micro' can fly at 2m/s with flapping frequency of 28Hz and 36° angle. By increasing the frequency to 34Hz 'Delfly Micro' is able to fly up to 3m/s with angle of 25°, depicted in Fig. (6(Middle)).

In view of the lift and thrust data plotted in Fig. (6), the 'Delfly Micro' can fly up to 3 m/s. In order to determine the required motor performance, the power consumption versus flapping frequency at different angle of attack is plotted in Fig. (7), for U=2 m/s. Noted that at the operation point around 30Hz, the motor should be powerful enough to produce 1 Watt power. In the design stage, the driving system including motor, controller, receiver and battery should be able to output 2 Watts power for reliable flight.

http://www.imavs.org/

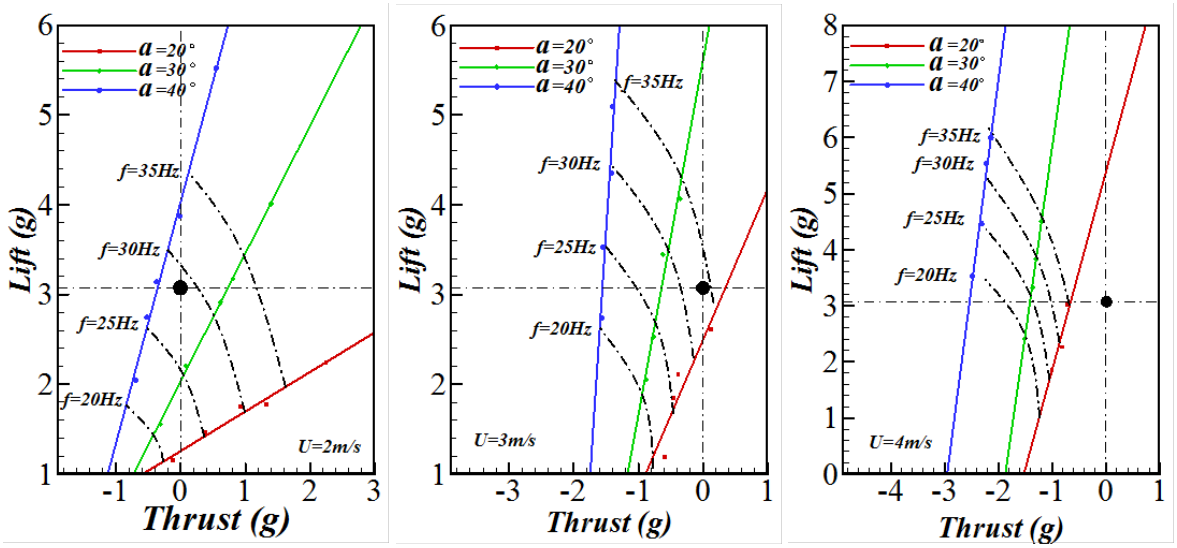


Figure 6: Lift and Thrust with varying incoming flow velocity. Left: U=2m/s; Middle: U=3m/s; Right U=4m/s.

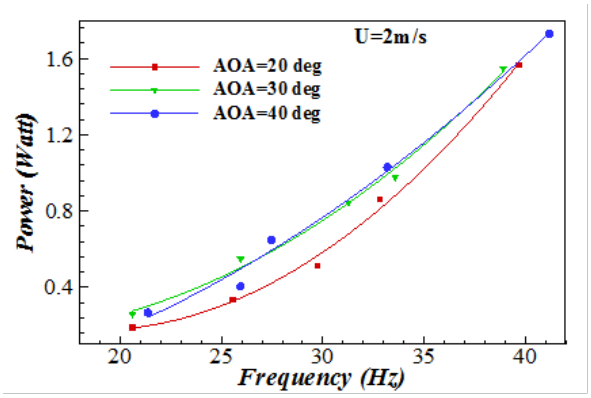


Figure 7: Power Consumption versus Frequency, U=2m/s.

4 Conclusion

There are two thrust peaks in one flapping period, and the peaks' location shifts with respect to flexibility and flapping frequency. The flexibility of the wing significantly influences the flapping performance. More rigid wings can produce more forces during flapping, however, they are more energy consuming. The standard wing built by 2m Mylar membrane possesses the best performance of the investigated wings, with slight scarifying in thrust compared with the wings made from 5 m and 10 m thickness Mylar. Wind tunnel tests define the initial flight envelop, and suggest that the 'Delfly Micro' can guarantee a level flight around 3m/s with 34Hz flapping frequency and 25° angle

http://www.imavs.org/

of attack. Based on the results, the calibrated driving system on 'Delfly Micro' should be powerful enough to produce 2 Watts power in further optimization. Further testing will be performed on the instantaneous deformation of the wing. Also the interaction between flapping wings and the tail has to be investigated for optimizing the tail position and mount angle.

Acknowledgement

This work has been supported by the Netherlands Technology Foundation (STW, project number 11023) and the China Scholarship Council (CSC). Thanks to the technical assistance from Stefan Bernardy in the Low Speed Lab of Delft University of Technology.

References

[1] G C H E de Croon, M A Groen, C De Wagter, B Remes, R Ruijsink, and B W van Oudheusden. Design, aerodynamics and autonomy of the delfly. *Bioinspiration and Biomimetics*, 7(2):025003, 2012.

[2] M. Percin Y. Hu B. W. van Oudheusden B. Remes and F. Scarano. Wing flexibility effects in clap-and-fling. *International Journal of Micro Air Vehicles*, 3(4):217–228, 2011.

[3] M. Percin M.Eisma B. W. van Oudheusden B. Remes R.Ruijsink and R. deWagter. Flow visulation in the wake of the flapping-wing mav 'delfly ii' in forward flight. *at the 30th AIAA Applied Aerodynamics Coference*, pages AIAA–2012–2664, 2012.

[4] W.Tay H.Bijl and B. W. van Oudheusden. Analysis of biplane flapping flight with tail. *at the 30th AIAA Applied Aerodynamics Coference*, pages AIAA–2012–2968, 2012.

[5] B.Bruggeman. Improving flight preformance of delfly ii in hover by improving wing design and driving mechanism. Master's thesis, Faculty of Aerospace Engineering, Delft University of Technology, 2010.

[6] T. Weis-Fogh. Quick estimates of flight fitness in hovering animals, including novel mechanisms for lift production. *Journal of Experimental Biology*, 59(1), 1973.

[7] J.Eisma. Flow visualization and force measurements on a flapping-wing mav delfly ii in forward flight configuration. Master's thesis, Faculty of Aerospace Engineering, Delft University of Technology, 2012.

http://www.imavs.org/

Resonance Based Flapping Wing Micro Air Vehicle.

Johannes F.L. Goosen[1], Hugo J. Peters[12], Qi Wang[1], Paolo Tiso[1] and Fred van Keulen[1].

[1]Delft University of Technology, Delft, The Netherlands
[2]Devlab, Eindhoven, The Netherlands
j.f.l.goosen@tudelft.nl

Abstract

Flapping wing micro air vehicles have many possible applications and have therefore been the subject of extensive research. The Atalanta project aims to develop a fully autonomous, 10 cm wingspan, flapping wing sensor platform. The structure is based on a fully compliant resonating structure with passive wing pitching. Several design improvements have taken place since the last report. Introduction of non-linear springs result in a more effective wing kinematic and extensive simulation and optimization was done to improve the wing design. Lighter electromagnetic actuators were designed while the possibilities of a chemical micro engine are investigated. Control experiments are underway using piezoelectric elements to locally change the structure stiffness thereby changing the kinematics of the system and thus the lift. To be able to efficiently design such a flight control, a framework was developed for the control of resonant compliant systems. Finally a sensor system for flight stabilization and object avoidance is suggested and under investigation.

http://www.imavs.org/

1 Introduction

Flapping wing micro air vehicles (FWMAV) have been under development for some time as a mobile platform for sensing and surveillance for applications such a search and rescue, measurement and mapping of dangerous or inaccessible places, and as mobile communication nodes. The flapping wing method of flight is particularly promising for indoor applications and small scale vehicles [1],[2], but it presents a number of unique challenges. Therefore, such a vehicle presents a fascinating research subject as many problems need to be solved before an autonomous flying FWMAV can be realized.

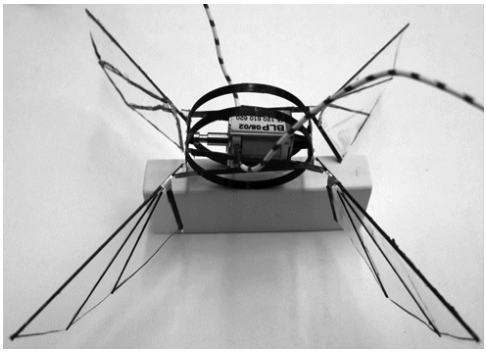


Figure 1: The first incarnation of the Atalanta FWMAV.

The Atalanta project was initiated as such a research vehicle by the industrial partners and aims at the development of a fully autonomous flapping wing flying sensor platform capable of hovering and slow flight with a wingspan of 10 cm. This size was chosen as it exhibits all the major problems without taking experiments and prototypes beyond normal manufacturing capabilities. However, all design decisions are made keeping further miniaturization in mind. Although the programme includes all issues from aerodynamics and flight kinematics, to control and navigation, most of the effort has been on the mechanical aspects, in particular wing flapping and kinematics. Figure 1 shows the design that was reported on previously [3]. Its four wings are driven by bringing the structure into resonance. The figure also shows the solenoid used in previous experiments to drive the structure.

2 Flight Kinematics

When looking at the performance of the MAV, the motion of the wings is the most important, next to weight. It is this wing motion that creates the aerodynamic forces and thus the lift and steering forces needed for flight. When considering the kinematics we can distinguish the basic sweeping motion of the wings and the pitching and deformation of each individual wing. Many papers have been published on the wing kinematics of insects [4][5][6], simulations of flapping wing motions using analytical models [7][8] and CFD [9][10], and experimental investigations [11][12]. However, the simulations and experiments usually simplify the problem to harmonic motions and the wing is assumed to be a plate. When looking at the flight of insects and hummingbirds we see quite different kinematics, which presents us with the question what kinematics would be desirable for a FWMAV and how to achieve this.

2.1 Wing Shape and Kinematics

The main consideration in the kinematic design is the motion of the wings. When we look at insects with a wingspan similar to the desired design, such as the hawkmoth (Manduca sexta), we see wing flapping that is not sinusoidal and includes an out of plane motion known as the heaving motion. Also torsion of the wing is apparent, resulting in varying angles of attack along the wing span as well as camber. These factors are usually not included in the design of FWMAVs, but their influence on wing performance is worth considering. To this end, an extensive investigation has been started on the optimization of wing shape and kinematics. The first results, using a quasi-steady aerodynamic model [13] and harmonic motions, show that many wing planforms have almost the same performance when considering lift production and efficiency [14][15]. A graph of the lift versus power performance of many optimized wings is shown in Figure 2 with four examples of corresponding wing shapes. In addition, this optimization shows the importance of wing torsion for efficient lift production and how this relates to wing shape. Preliminary investigation of wing performance with respect to heaving and torsion show the influence of both in particular for passive wing pitching, and further optimization will result in a more optimal wing shape, along with the corresponding kinematic. Investigation of insect wing kinematics often shows a distinctly non-sinusoidal wing stroke kinematic [14][6], and simulations with both the quasi-steady model and CFD have shown that a more constant wing translation and quick reversal at the end of the stroke increases lift over the sinusoidal kinematic [16].

http://www.imavs.org/

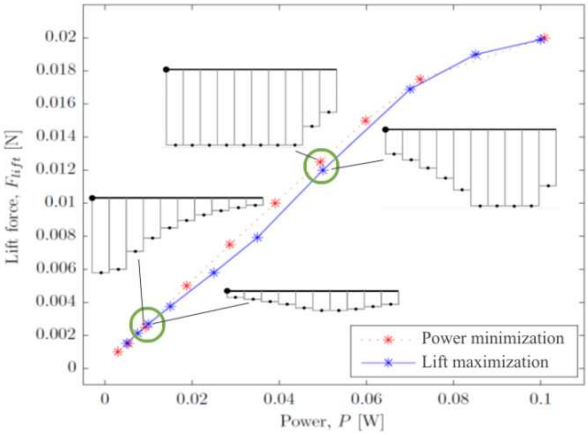


Figure 2: Optimized wing planforms for different lift and power requirements showing very different wing shapes with almost identical performance.

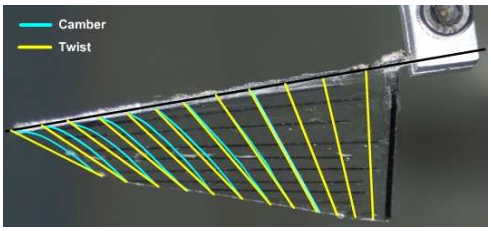


Figure 3: Front view of an experimental wing at midstroke showing pitching, twist and camber.

Practical implementation of this kinematic in a FWMAV is not trivial as full control of all degrees of freedom would require complex mechanics and several actuators per wing. However, insects manage to create their wing kinematic using passive wing pitching and deformation. A similar approach would do away with the need for full control over all kinematic parameters and use just one or two [16]. Simulations show that passive pitching and torsion can achieve efficient results that approach the optimal kinematic but require a heaving motion. Implementation of such performance in a wing is difficult, but experimental results are promising and have shown the desired pitching and twist. Figure 3 clearly shows the pitching and twist in a the front view of a wing in motion. It is also clear that this wing shows considerable camber which has not been included in the simulations. Wing design and the influence of camber is the subject of further research.

2.2 Resonance Based Flapping

An issue often discussed in insect inspired flight is that of resonance based kinematics for the wing flapping motion. Most insects seem to make use of resonance to drive the main flapping motion [18][19]. Although the advantages, such as the cancellation of losses due to inertia and the decoupling of aerodynamic loading and actuation are clear, hardly any of the current FWMAV designs use resonance. The advantages of such a system are countered by several difficulties.

- Actuation. A resonant system requires an actuator that is force coupled to the system instead of the more common displacement coupling. The most obvious solution is a linear electromagnetic actuator, but this limits the actuation frequency to the resonance (flapping) frequency, which introduces problems with the power needed per cycle.
- Flight control. Changing the flapping frequency, which is an effective way of controlling lift is not an option. Varying the drive frequency has a negative impact on the flapping amplitude and therefore the lift production of the wings.
- Weight. In order to have any resonance, a quality factor of more than 2 is needed which requires considerable energy storage in such a highly damped system. For this we need a more stiff spring structure, which will add weight to the FWMAV.

http://www.imavs.org/

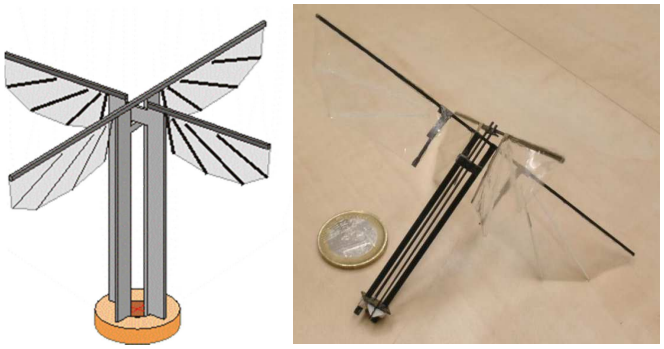


Figure 4: Schematic (left) and realization (right) of the four wing vibrating design using torsion springs for non-linear spring stiffness, resulting in better flapping kinematics.

To further investigate the possibilities of resonance, the Atalanta was designed as a resonant compliant structure [3]. The previously reported configuration consists of a ring shaped spring connected to a compliant mechanism to transform the linear motion into the required wing rotation. The overall system is designed with a resonance frequency of approximately 30 Hz, which is reported to be the most efficient for the 5 cm wings [20], and a quality factor of approximately 2.5 in air. Due to the linear spring behaviour of the rings the flapping motion of the wings is sinusoidal. As was discussed earlier, non-sinusoidal motion seems preferable and can be achieved in a resonant system using non-linear or bi-stable mechanisms. The use of non-linear springs or pre-tension can be used to convert the sinusoidal displacement into a more triangular motion. Although this allows for a more advantageous kinematic it can complicate the structure and has the disadvantage that the vibration frequency of such a system becomes amplitude dependent. To experiment with this non-linear behaviour a different structure was designed that also simplifies manufacture and makes the structure more robust. The new design is shown in Figure 4 and makes use of two counter rotating torsional springs. The rotation allows for a direct connection between the wing and the spring and does away with the conversion mechanism. The configuration of the torsion springs in combination with the large rotation results in the non-linear response and a more constant wing translation velocity and faster wing reversal. Further optimization of this non-linear spring is underway, as is the use of both pre-tension and bi-stability to achieve an optimal wing sweeping kinematic. For more efficient flight when using passive pitching a heaving motion is desired resulting in the figure eight motion of the wingtip. Like the pitching, we would like to achieve this in a passive manner without the need for actuators or control. The first design introduces some torsion flexibility close to the root of the flat spar, which, with the moment caused by the passive pitching results in a torsion and out of plane bending of the wing spar, which gives the desired heaving motion.

3 Actuation

Actuation is one of the biggest challenges of FWMAVs. Driving a resonant based structure is difficult as the design requires reciprocal motions which are limited to the vibration frequency of the design. Two options have been under investigation. First, electromagnetic actuators, as these are easy to control and make. Although, the energy density of such actuators is relatively low [21] when taking into account their efficiency and additional requirements with respect to drive circuitry and comparing them to other options at actuation frequencies of 30 Hz, they compare quite favourably. Two electromagnetic actuators were designed. The first

http://www.imavs.org/

linear design consists of a simple coil and two permanent magnets, which is used to drive the ring shaped spring of the original design. This design outperforms more sophisticated Lorenz force actuators due to the low volume of the relatively heavy core material for the magnetic circuit. The strong asymmetry of the force output of the actuator is compensated partially by modification of the spring but additional non-linearity is of little importance due to the resonant nature of the system. The second actuator uses a permanent magnet that can rotate in a coil and thus will try to align itself with the field. This actuator is used to drive the torsional spring of the new design.

As an alternative, the feasibility of a chemical engine is under investigation. Such an engine can take advantage of the high energy density of a chemical fuel and the reduction in weight as the fuel is used. Previous research on a suitable fuel and reaction for such an engine showed the potential of the pulsed catalytic decomposition of hydrogen peroxide. However, the engine at the scale required does not exist. In order to design such an engine, an extensive thermodynamic investigation of the primary methods for converting a chemical reaction into mechanical work was done, taking the problems of miniaturization into account. When considering the application, which requires a reciprocal motion, the Otto engine principle was found to be the most promising. Research is underway on the design of such a miniature chemical engine for FWMAV applications.

http://www.imavs.org/

4 Flight control

In order to stabilize and steer the FWMAV, control is required. In a resonating structure with passive wing pitching, no direct handles are available for the control of the vehicle as structures such as a tail and rudder exclusively for control purposes are undesirable due to their weight. In order to control the vehicle, the basic symmetry of lift production of the four wings must be broken in order to create a rolling motion which is required for stabilizing hover and through a tilt in the thrust vector allows for sideway motion. As the whole structure can be considered a resonating system, the kinematic of the system is determined by the stiffness and mass distribution in this system. The lift production of each wing during flight is determined by this kinematic, and any changes in this distribution will result in a change of kinematics and therefore in lift production. In order to determine where to change the structure to achieve the desired kinematic response with the least amount of energy, a general design framework was developed based on using the modal description of the structure, required modification and results of actuation [22][23]. This allows for an optimization of the placement of control actuators to achieve the needed controls.

The structural changes caused by the control actuators can include variations in stiffness, damping or redistribution of mass. Of these, changes in stiffness and damping are the simplest to achieve through the use of piezoelectric patches. Passive control through the opening or closing of the contact between the electrodes will result in a change of apparent stiffness of the patch without the need for additional energy. Simulations have shown that local stiffness changes up to 20% are possible, which is sufficient to achieve the kinematic changes and therefore the control required for flight stabilization and steering. Experiments are underway to determine the change in kinematic that can be achieved using such a system.

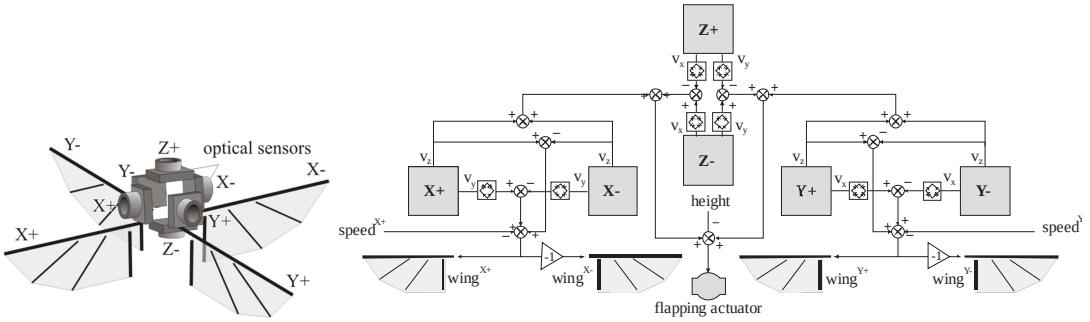


Figure 5: Stabilization and object avoidance system for hovering and forward flight. Placement of the optical sensors (left) and a possible analogue control system (right).

Apart from the changes that can be achieved this way, the initial kinematics and how this produces lift is of great importance to the control. The wings shape and kinematic determine where the resultant force is with respect to the centre of mass and therefore will have great influence on how a change of lift production relates to a control moment. Consider the two top wings in Figure 2. Both produce virtually identical lift and require the same power in hovering flight. However, this resulting lift acts closer to the wing root for the wing on the left than for the wing on the right, resulting in a different moment around the centre of mass. This shows that de control requirements should be taken into account when designing the wings.

5 Sensors and Control Systems

For the stabilization of the FWMAV and autonomous flight, sensors are of the utmost importance as the time constants for rolls and disturbances are so small that a pilot can not correct them. In addition, the possibility of autonomous flight must be kept in mind as the ultimate goal. The requirements of weight and power limit the possible sensor and control systems that can be used. Inspired by insects senses, a sensor and control system is under investigation that uses image flow sensors and a simple control circuit. Similar systems have been shown to be effective for stabilization and obstacle avoidance in fixed wing MAVs [24],[25] but have not been applied successfully to hovering vehicles. However, these system are attractive in that they used simple analogue control systems and can be designed to show object avoidance behaviour without the need for complex image processing. A flow sensor based control system for flight stabilization and object avoidance was designed (see Figure 5) [26], and is now under investigation.

6 Conclusions

The FWMAV developed within the Atalanta project has shown the possibilities of using resonant systems for the basic wing sweeping and passive pitching of the wings. The investigation of the wing design space has shown that considerable advances in lift and efficiency can be realized by using the proper wing shape and wing kinematic. This includes the passive pitching, twisting and heaving of the wing. It also shows that control considerations play a role in most aspects of the design. Electromagnetic actuators where designed to drive the two FWMAVs but actuation and power storage are still the main challenge in the development of a usable FWMAV. Further research will continue in the design of chemical micro engines. To control the FWMAV, piezoelectric patches placed around

http://www.imavs.org/

the structure can be used without the need for additional power, high voltages and at a minimal additional weight. However, the sensor and control systems required for stabilization and autonomous flight have not been realized and require further research.

The design of FWMAVs cannot be broken down in the design of separate subsystems. An overall design approach is necessary for achieving autonomous flight and unconventional solutions, such as resonant compliant structures, non-linear mechanics and analogue controls systems must be considered and developed further.

Acknowledgements

The authors would like to thank the Development Laboratories, Eindhoven, the Netherlands for initiating the project and their continued support, and the many master and bachelor students that worked on the Atalanta project over the past years. We would also like to thank Point-one for the financial support of the project 'Control of Resonant Compliant Systems' (RRCS) and the Chinese Research Council for support of the project on 'Design and optimization of wings for a flapping-wing micro air vehicle'.

http://www.imavs.org/

References

[1] M.H. Dickinson, F.O. Lehmann, S.P. Sane, Wing rotation and the aerodynamic basis of insect flight. *Science* 284(5422): 1954-1960, 1999.

[2] C.P. Ellington, The novel aerodynamics of insect flight: Applications to micro-air vehicles. *Journal of Experimental Biology* 202: 3439-3448, 1999.

[3] C.T. Bolsman, J.F.L. Goosen, and F. Van Keulen. Design overview of a resonant wing actuation mechanism for application in flapping wing mavs. *International Journal of Micro Air Vehicles*, 1(4):263–272, 2009.

[4] C. P. Ellington. The Aerodynamics of Hovering Insect Flight. III. Kinematics. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 305(1122):145–181, 1984.

[5] AP Willmott and CP Ellington, The mechanics of flight in the hawkmoth Manduca sexta. I. Kinematics of hovering and forward flight. *Journal of Experimental Biology*, 200(21):2705–2722, 1997.

[6] Y.H. Chen, M. Skote, Y. Zhao, and W.M. Huang. Dragonfly (Sympetrum flaveolum) flight: Kinematic measurement and modelling. *Journal of Fluids and Structures*, 2013.

[7] S. P. Sane 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):1087–1096, 2002.

[8] Z. J. Wang. Unsteady forces and flows in low Reynolds number hovering flight: two-dimensional computations vs robotic wing experiments. *Journal of Experimental Biology*, 207(3):449–460, 2004.

[9] H. Aono, Chang-kwon Kang, Carlos E. S. Cesnik, and Wei Shyy. A numerical framework for isotropic and anisotropic flexible flapping wing aerodynamics and aeroelasticity. In *28th AIAA Applied Aerodynamics Conference*, 1–25, Chicago, Illinois, 2010.

[10] T. Gillebaart, A.H. van Zuijlen, and H. Bijl, Aerodynamic analysis of the wing flexibility and the clap-and-peel motion of the hovering DelFly II, *International Micro Air Vehicles conference* 2011.

[11] B. Singh, M. Ramasamy, I. Chopra, J. G. Leishman, Experimental studies on insect-based flapping wings for micro hovering air vehicles, *American Institute of Aeronautics and Astronautics*, 20742, 2004.

[12] M. Percin, Y. Hu, B.W. van Oudheusden, B. Remes and F. Scarano, Wing flexibility effects in clap-and-fling, *International Micro Air Vehicles conference* 2011.

[13] Gordon J. Berman and Z. Jane Wang. Energy-minimizing kinematics in hovering insect flight. *Journal of Fluid Mechanics*, 582:153, 2007.

[14] Lehmann, F. O. and M. H. Dickinson, The control of wing kinematics and flight forces in fruit flies (Drosophila spp.). *J. Exp. Biol.* 201(3): 385-401. 1998.

[15] Hugo J. Peters, Johannes F.L. Goosen, and Fred van Keulen. Flapping wing performance related to wing planform and wing kinematics. In *12th AIAA Aviation Technology, Integration, and Operations* (ATIO) *Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization*, Aviation Technology, Integration, and Operations (ATIO) Conferences. American Institute of Aeronautics and Astronautics, 2012.

[16] H. Dong, Z. Liang and M. Harff, Optimal Settings of Aerodynamic Performance Parameters in Hovering Flight, *International Journal of Micro Air Vehicles*, 1 (3), 173-181, 2009.

[17] A.R. Ennos. The inertial cause of wing rotation in Diptera. *Journal of Experimental Biology* 140:161–169, 1988.

[18] C. H. Greenewalt. The wings of insects and birds as mechanical oscillators. *Proceedings of the American Philosophical Society*, 104(6):605–611, 1960.

[19] T. Weis-Fogh, Quick estimates of flight fitness in hovering animals, including novel mechanisms for lift production, *Experimental Biology* 59 (1), 169-230.

[20] C. Berg and J. Rayner, "The moment of inertia of bird wings and the inertial power requirement for flapping flight," *J. Exp. Biology*, 198 (8), 1655–1664, 1995.

[21] Michael Karpelson, Gu-Yeon Wei, Robert J. Wood, A Review of Actuation and Power Electronics Options for Flapping-Wing Robotic Insects, In *2008 IEEE International Conference on Robotics and Automation Pasadena*, CA, USA, 2008.

[22] H.J. Peters, P. Tiso, J.F.L. Goosen, F. van Keulen; Control of resonating fwmav structures using repeated eigenvalues, In: *International Forum on Aeroelasticity & Structural Dynamics 2013* 24-26 June 2013 - Bristol UK.

[23] H.J. Peters, J.F.L. Goosen, P. Tiso, F. van Keulen; Control of Harmonically Driven Resonating Compliant Structures using Local Structural Modifications; In: *ECCOMAS Multibody Dynamics 2013*, Z. Terze, M. Vrdoljak (ed.) ISBN: 978-953-7738-22-8 01-04, July 2013 - Zagreb Croatia.

[24] A. Beyeler, J.-Ch. Zufferey D. Floreano, optiPilot: control of take-off and landing using optic flow, in *European Micro Air Vehicle conference and competition 2009,* EM2009.

[25] Mahdi Rezaei and Fariborz Saghafi, Optical flow-based obstacle avoidance of a fixed-wing MAV, *Aircraft Engineering and Aerospace Technology: An International Journal,* 83(2), 85–93, 2011.

[26] J.F.L. Goosen, Design aspects of a bio-inspired flying sensor node, In *IEEE Sensors 2012*, Taiwan, 2012.

http://www.imavs.org/

Technical Session 6

System, navigation and control

Chairman : Gautier Hattenberger

- Montserrat Manubens, Didier Devaurs, Lluis Ros and Juan Cortés. ***A motion planning approach to 6-D manipulation with aerial towed-cable systems***
- Karl Kufieta and Peter Vörsmann. ***First flight experiments with a RT-Linux autocode environment including a navigation filter and a spline controller***
- Bart Remes, Dino Hensen, Christophe De Wagter, Erik van der Horst and Guido De Croon. ***Paparazzi open source autopilot adapted for use with the Parrot AR drone***

http://www.imavs.org/

A motion planning approach to 6-D manipulation with aerial towed-cable systems

Montserrat Manubens[1], Didier Devaurs[2,3], Lluís Ros[1] and Juan Cortés[2,3]

[1] Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain
{mmanuben, lros}@iri.upc.edu
[2] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[3] Univ de Toulouse, LAAS, F-31400 Toulouse, France
{ddevaurs, jcortes}@laas.fr

Abstract

We propose a new approach for the reliable 6-dimensional quasi-static manipulation with aerial towed-cable systems. The novelty of this approach lies in the combination of results deriving from the static analysis of cable-driven manipulators with a cost-based motion-planning algorithm to solve manipulation queries. Such a combination of methods is able to produce feasible paths that do not approach dangerous/uncontrollable configurations of the system. As part of our approach, we also propose an original system that we name the *FlyCrane*. It consists of a platform attached to three flying robots using six fixed-length cables. Results of simulations on 6-D quasi-static manipulation problems show the interest of the method.

http://www.imavs.org/

1 Introduction

Aerial towed-cable systems have been used for decades, mainly as crane devices. They have proved to be very useful in various contexts, such as supply delivery missions and rescue operations [3], as well as environmental monitoring and surveillance [16]. One such system has even been successful as a safe soft-landing device for a rover on the martian surface [15], for instance. In all these examples, the systems only required a certain position accuracy, for example to execute simple trajectories [14, 13]. Little work has been done on trying to govern a load in both position and orientation. To the best of our knowledge, the only existing technique for 6-dimensional manipulation with an aerial towed-cable system requires a given discrete set of load poses [12, 6]. Such a technique relies on solving the inverse kinematics problem and determining the static equilibrium for all given poses. Requiring a given set of platform poses may be too restrictive, though, especially in constrained workspaces, because it may provide no result, while there may exist solutions for other intermediate poses.

We have recently developed a new reliable motion planning approach for 6-dimensional quasi-static manipulation with aerial towed-cable systems [11] [1]. The proposed method only requires a start and goal configurations as input, and provides a feasible path to achieve the manipulation task. In addition to being feasible, the generated manipulation path will be of *good quality*, meaning that all intermediate configurations will fulfill adequate physical properties related to the forces applied to the system and to the cable tensions. This quality will be measured by a formal criterion derived from the static analysis of the system, based on a similar formulation as that used for cable-driven manipulators [5, 4]. A path-planing algorithm taking this quality measure into account [8] will then be applied to compute good-quality paths.

In addition to the methodology, this paper presents an aerial towed-cable system to perform 6-D manipulation tasks, that we call the *FlyCrane*. This system consists of a moving platform attached to three flying robots by means of six fixed-length cables linked by pairs to each robot. The 6-D manipulation of the platform

[1] This paper is a short version of the original publication [11], which should be consulted for additional details on the method.

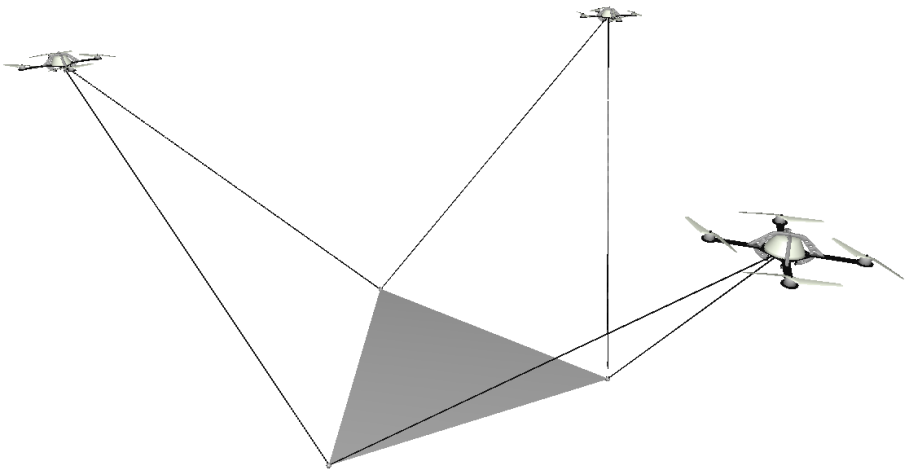


Figure 1: Octahedral version of the *FlyCrane* system.

can be performed by varying the relative positions of the flying robots. An octahedral version of this system is illustrated in Figure 1.

The rest of the paper is organized as follows: Section 2 provides an overview of our approach, detailed in [11]. Section 3 presents an evaluation of our approach on two 6-D manipulation problems involving the octahedral version of the *FlyCrane* system. Section 4 provides conclusions and discusses possible directions for future work.

2 Overview of the approach

Towed-cable systems present important analogies with cable-driven manipulators, which enable us to perform their static analysis in a way similar to that presented in [4]. However, while cable-driven manipulators have to adjust the lengths of their cables to reach a precise pose of the platform, towed-cable systems have fixed-length cables and are actuated by displacing their anchor points. Manipulating the six degrees of freedom of a load requires a minimum of seven cables, unless some convenient forces reduce this number. In crane configurations, for instance, gravity acts as an implicit cable, and therefore six cables suffice for the full 6-D manipulation. Examples of such structures are the NIST Robocrane [1] or more general cable-driven hexapods [4].

In the proposed aerial towed-cable system, called the *FlyCrane*, the platform is also pulled by six cables, which, as illustrated in Figure 1, are pairwise attached to three flying robots (instead of being individually attached to six flying robots). It is worth noting that three is the minimal number of flying robots required to properly operate this system, as less robots would not allow the manipulation of the six degrees of freedom of the platform. Whenever the cable base points are also coupled, we call it *octahedral FlyCrane*, because the structure can be seen as an octahedron, comprising the following 8 triangles: the platform base points, the triangle formed by the flying robots, and the 6 triangles made of pairs of adjacent cables.

In this paper we assume that motions are performed quasi-statically, thus neglecting the dynamic analysis of the system. Although it may appear as a strong simplification, this assumption is frequently made in fine-positioning situations, where slow motion is imperative. Nevertheless, dealing with dynamical aspects can be an interesting extension for future work, as will be discussed in Section 4.

http://www.imavs.org/

Even with six cables, the six degrees of freedom of the platform can be governed only in a subset of the configuration space of the system. Indeed, the pose of the platform is locally determined only when all cables are in tension. Therefore, it is important to prevent the cables from being slack or too tight. Besides, the flying robots must be able to counteract the forces exerted on them. These two conditions determine the feasibility of a configuration of the system. More precisely, to be feasible, a configuration must satisfy the following two types of constraints:

- *Wrench-feasibility* constraints: they guarantee that the system is able to statically counteract a set of wrenches applied on the platform while ensuring that the cable tensions always lie within a pre-defined, positive acceptance range; they are derived from the static analysis of cable-driven manipulators [5, 4].
- *Thrust* constraints: they guarantee that the thrust of the flying robots can equilibrate the forces applied on them, namely the forces exerted by the cables and the force of gravity.

The set of configurations that satisfy these feasibility constraints form a manifold in which paths have to be searched. An infinite number of feasible solution paths may exist for a given manipulation query. A way to discriminate the less appropriate ones is to define a criterion assessing their quality. A good-quality path should be a path whose intermediate configurations are attributed a low cost with respect to the physical properties of the system. A meaningful way to evaluate the cost of a configuration of the system is to derive it from the previous feasibility constraints. The idea is to define a cost function that tends to infinity when a configuration approaches the limit of a feasibility constraint (i.e. when a cable tension approaches one of its limits or when a robot approaches its maximum thrust) and that takes low positive values when a configuration is far from the non-feasible ones. Such a cost function can be written as a combination of terms from the equations that define the feasibility conditions. In addition, it is possible to show that it is a continuous differentiable function over the set of feasible configurations, which is a crucial property for a suitable performance of the path planning method applied next.

Any general path planner, such as the Rapidly-exploring Random Tree (RRT) algorithm [10], could be applied to compute collision-free paths satisfying the aforementioned feasibility constraints to perform 6-D manipulation tasks with the *FlyCrane* system. However, it might not produce good-quality paths. Since we are able to define a cost function over the configuration space, we can use a cost-based path planner, such as the Transition-based RRT (T-RRT) [8], in order to obtain good-quality manipulation paths. T-RRT has been successfully applied to various types of problems in robotics [8, 2] and structural biology [9]. Nevertheless, it is worth noting that, to the best of our knowledge, this is the first time it is applied to aerial manipulation problems.

3 Results

In this section, we evaluate the proposed approach on two 6-D quasi-static manipulation problems involving the *FlyCrane* system (cf. Fig. 1). The first example is a complex task (inspired by classical motion planning benchmarks) in which the *FlyCrane* has to get a 3-D puzzle piece through a hole, as illustrated by Fig. 2. The second example, presented in Fig. 3, simulates a more realistic situation in which the *FlyCrane* has to install a lightweight footbridge between two buildings to evacuate people during a rescue operation. These examples differ in terms of difficulty: the *Rescue* problem is the easiest one because the manipulation task involves translation and rotation about a single axis, whereas the *Puzzle* problem requires a complex coordinated motion, with simultaneous translation and 3-D rotation of the platform.

On both examples, we evaluate the performance of the RRT and T-RRT algorithms on the basis of their running time t (in seconds), the number of attempted expansions X, and the number of nodes N in the produced tree. To avoid generating trivially-non-feasible paths, RRT only accepts collision-free configurations that

http://www.imavs.org/

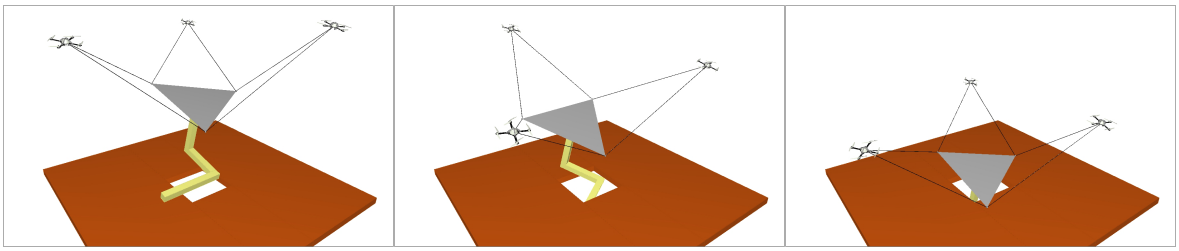


Figure 2: The *Puzzle* problem: the *FlyCrane* has to get a 3D puzzle piece through a hole.

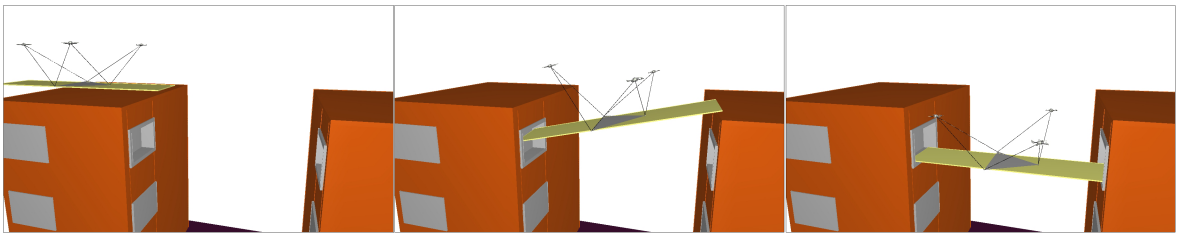


Figure 3: The *Rescue* problem: the *FlyCrane* has to install a lightweight footbridge between two buildings for a rescue operation.

http://www.imavs.org/

satisfy the aforementioned wrench and thrust feasibility constraints. After performing a smoothing operation (based on the random shortcut method [7]) on the paths generated by RRT and T-RRT, we evaluate the path quality by computing the average cost $avgC$, the maximal cost $maxC$, the mechanical work MW, and the integral of the cost IC. The *mechanical work* of a path is defined as the sum of the positive cost variations along the path [8]. Table 1 reports values for all variables averaged over 100 runs of the algorithm.

Unsurprisingly, Table 1 shows that T-RRT provides better-quality paths than RRT on both examples: on the *Puzzle* problem, all cost statistics are more than one order of magnitude lower for paths generated by T-RRT; on the *Rescue* problem, they are between three and 50 times lower. Since it generally requires more expansion attempts to find configurations with acceptable cost, T-RRT is often slower than RRT, as is the case on the

Table 1: Evaluation of RRT and T-RRT on the *Puzzle* and *Rescue* problems. Average values over 100 runs are given for: the average cost $avgC$, the maximal cost $maxC$, the mechanical work MW, the integral of the cost IC, the running time t (sec.), the number of nodes N in the tree, and the number of expansion attempts X.

	Puzzle						
	$avgC$	$maxC$	MW	IC	t (s)	N	X
RRT	1130	11,684	11,651	300,793	34	2654	15,609
T-RRT	78	229	193	30,352	169	4698	78,501

	Rescue						
	$avgC$	$maxC$	MW	IC	t (s)	N	X
RRT	102	575	554	80,750	126	1361	193,517
T-RRT	36	42	11	24,588	54	379	207,778

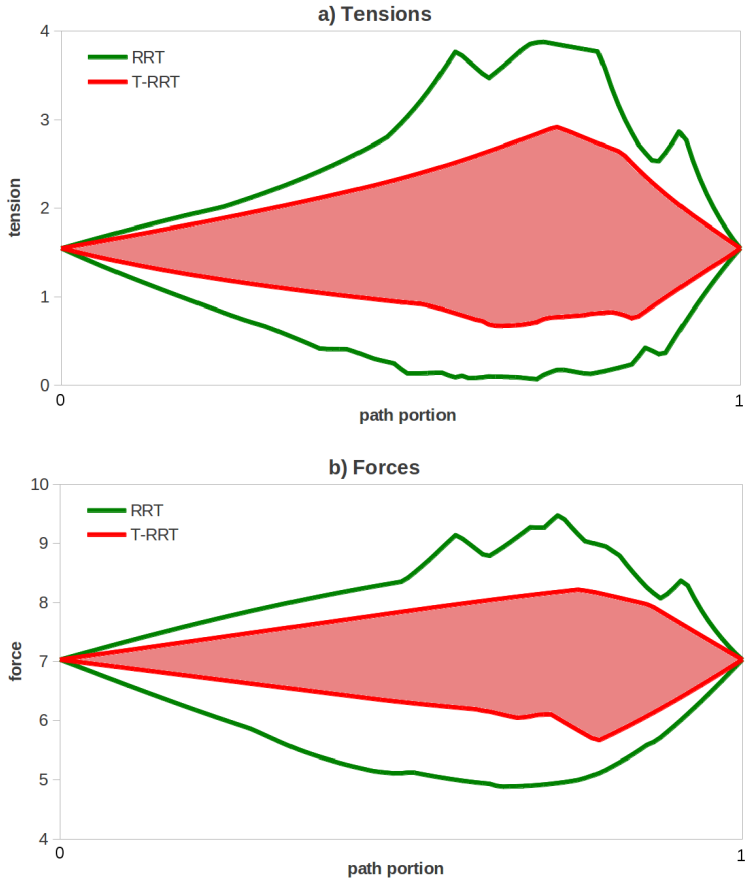


Figure 4: Profiles of a) the tension range and b) the force range, observed over 100 paths produced by RRT and T-RRT on the *Rescue* problem. The filled areas between the red curves represent the ranges for T-RRT; the areas between the green curves represent the ranges for RRT.

Puzzle problem (169 s vs 34 s). However, it is worth noting that T-RRT runs faster than RRT on the *Rescue* problem (54 s vs 126 s), thanks to the lower number of nodes added to the tree (379 vs 1361), which makes the nearest-neighbor search faster.

We were interested in finding out what made path quality differ between RRT and T-RRT. For that, we computed the tensions exerted on each cable and the forces exerted on each quadrotor, along the paths produced by RRT and T-RRT, after dividing every path into 100 steps corresponding to intermediate configurations of the system. Then, for each path-step, we computed the minimal and maximal tensions (over all cables) and forces (over all quadrotors) over the 100 paths produced by RRT and over the 100 paths produced by T-RRT. Therefore, for each step, we obtained the tension ranges and the force ranges yielded by RRT and T-RRT. Fig. 4 presents the profiles of the tension range and of the force range, respectively, on the *Rescue* problem. Similar plots have been obtained on the *Puzzle* problem. We can see that using T-RRT leads to smaller tension and force ranges than using RRT. Most importantly, we observe that RRT produces paths along which a tension or a force can be dangerously close to a bound of its validity interval. For example, Fig. 4.a shows that, along some path, at least one tension comes close to zero, meaning that at least one cable almost goes slack. Similarly, on the

http://www.imavs.org/

Puzzle problem, one force comes close to the maximal thrust value. As a conclusion, we argue that integrating the path-planning T-RRT algorithm into the proposed 6-D manipulation approach allows us to plan safer paths for the *FlyCrane* system.

4 Conclusion

We have presented an approach for the 6-dimensional quasi-static manipulation of a load with an aerial towed-cable system. The main contribution of the approach lies in the combination of results deriving from the static analysis of cable-driven manipulators with a cost-based motion-planning algorithm to solve manipulation queries. The link underlying this combination is the definition of a quality measure for the configurations of the system. First, this quality measure is based on the wrench-feasibility constraints applied to cable-driven manipulators and on additional thrust constraints, and allows: 1) to discriminate non-feasible from feasible configurations, and 2) to favor configurations that are far from violating these constraints, by attributing them a low cost. Second, this quality measure leads to the definition of a cost function, thus allowing for the use of a cost-based motion-planning algorithm, namely the Transition-based RRT (T-RRT). As a result, rather than simply computing collision-free paths, the proposed approach produces good-quality paths, with respect to the constraints imposed on the system.

As part of our approach, we have additionally proposed an aerial towed-cable system that we have named the *FlyCrane*. This system consists of a platform attached to three flying robots by means of three pairs of fixed-length cables. We have evaluated the approach, in simulation, on two 6-D manipulation problems involving an octahedral version of the *FlyCrane* system. The results of the evaluation show that the proposed motion planning approach is suitable to solve 6-D quasi-static manipulation tasks. Furthermore, they have confirmed that RRT, which is the original variant of T-RRT that does not take the cost into account, may produce paths that occasionally approach dangerous situations, while T-RRT produces safer paths.

The proposed approach allows for extensions in several ways. In particular, we expect to extend the method to consider positioning errors for the flying robots, which could be due to external force perturbations and to errors in the localization methods. Additionally, an interesting and challenging extension to this work is the introduction of dynamics in the motion of the load and of the flying robots, as they play an important role in the overall manipulation of the system.

In this paper, we have applied the proposed approach in simulated environments. As part of our future work, we plan to implement this approach in a real aerial towed-cable system. This will serve as a testbed for the validation of the method and its further extensions, providing relevant feedback on the real limitations of the approach and the system. In real-life situations, the proposed approach could be helpful in various applications. As illustrated by the simulated *Rescue* problem, one possible application is the construction of platforms for the evacuation of people in rescue operations. Another application could be the installation of platforms in uneven terrains for the landing of manned or unmanned aircrafts. More generally, it could be useful for the assembly of structures in places difficult to access for humans.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under contract DPI2010-18449, by the European Community under contract ICT 287617 “ARCAS”, and by a *Juan de la Cierva* contract supporting the first author.

http://www.imavs.org/

References

[1] J. Albus, R. Bostelman, and N. Dagalakis. The NIST Robocrane. *J. Robotic Systems*, 10(5):709–724, 1993.

[2] D. Berenson, T. Siméon, and S. Srinivasa. Addressing Cost-Space Chasms in Manipulation Planning. In *IEEE Int. Conf. Robotics and Automation*, pages 4561–4568, 2011.

[3] M. Bernard, K. Kondak, I. Maza, and A. Ollero. Autonomous transportation and deployment with aerial robots for search and rescue missions. *J. Field Robotics*, 28(6):914–931, 2011.

[4] O. Bohigas, M. Manubens, and L. Ros. Navigating the Wrench-Feasible C-Space of Cable-Driven Hexapods. In *Cable-Driven Parallel Robots*, pages 53–68. Springer, 2012.

[5] P. Bosscher, A.T. Riechel, and I. Ebert-Uphoff. Wrench-feasible workspace generation for cable-driven robots. *IEEE Trans. Robotics*, 22(5):890–902, 2006.

[6] J. Fink, N. Michael, S. Kim, and V. Kumar. Planning and control for cooperative manipulation and transportation with aerial robots. *Int. J. Robotics Research*, 30(3):324–334, 2011.

[7] R. Geraerts and M. H. Overmars. Creating high-quality paths for motion planning. *Int. J. Robotics Research*, 26(8):845–863, 2007.

[8] L. Jaillet, J. Cortés, and T. Siméon. Sampling-based path planning on configuration-space costmaps. *IEEE Trans. Robotics*, 26(4):635–646, 2010.

[9] L. Jaillet, F.J. Corcho, J.J. Pérez, and J. Cortés. Randomized tree construction algorithm to explore energy landscapes. *J. Computational Chemistry*, 32(16):3464–3474, 2011.

[10] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics: New Directions*, pages 293–308. A K Peters, 2001.

[11] M. Manubens, D. Devaurs, J. Cortés, and L. Ros. Motion planning for 6-D manipulation with aerial towed-cable systems. In *Proc. of Robotics: Science and Systems*, 2013.

[12] N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, 2011.

[13] R. M. Murray. Trajectory Generation for a Towed Cable System Using Differential Flatness. In *IFAC World Congress*, 1996.

[14] R. A. Skop and Y. I. Choo. The Configuration of a Cable Towed in a Circular Path. *J. Aircraft*, 8:856–862, 1971.

[15] A. Steltzner, D. Kipp, A. Chen, D. Burkhart, C. Guernsey, G. Mendeck, R. Mitcheltree, R. Powell, T. Rivellini, M. San Martin, and D. Way. Mars Science Laboratory entry, descent, and landing system. In *IEEE Aerospace Conf.*, 2006.

[16] P. Williams. Optimal terrain-following for towed-aerial-cable sensors. *Multibody System Dynamics*, 16 (4):351–374, 2006.

http://www.imavs.org/

UAV Flight Experiments with a RT-Linux Autocode Environment including a Navigation Filter and a Spline Controller.

Karl Kufieta[1], Christian Wickbold[1] and Prof. Vörsmann[1]

Institute of Aerospace Systems, TU Braunshschweig, Braunschweig, Germany
karl.kufieta@tu-braunschweig.de

Abstract

A common method for navigation in Unmanned Aerial Vehicles is the fusion of the GPS-position and an Inertial Measuring Unit. For this data fusion a Kalman Filter is used. To guide and control the Unmanned Aerial Vehicle, spline paths in combination with linear or nonlinear controllers are suitable. The development of these algorithms is iterated cyclically beginning with a simulation and then testing the algorithms on the hardware. To be able to repeat fast development cycles an autocode environment in combination with a real-time Linux has been developed. This paper describes the performance of the used autopilot architecture in the scope of algorithm development. Finally the recorded flight test data is compared to the simulation data.

http://www.imavs.org/

1 Introduction

Unmanned aerial vehicles (UAV) as depicted in fig. 1 are a testbed for a wide field of applications. Research at the ILR (institute of aerospace systems) ranges from navigation and control algorithms, over modeling and parameter identification to flight missions, e.g. meteorological [11] campaigns.



Figure 1: Left: Arctic mission with Unmanned Aerial Vehicle, Right: experimental Multiplex Twinstar with mounted autopilot

Today's microprocessors are powerful and suitable for complex algorithms. As those processors are small and energy efficient they can be used in small UAV's. Usually a processor is programmed in a high level language like C or C++. Complex algorithms for navigation and control of UAV's or robots in general are often programmed in object oriented simulation

environments like Matlab Simulink. In these environments signals are represented by lines between operators (e.g. plus, minus, transfer function), sources (e.g. constant, sensor) and sinks (display, actuator). To implement an algorithm from the simulation world, the vendors (like Mathworks) have designed code generators. With a code generator it is possible to translate the object oriented simulation into C-code. This process can be automated so far that it is possible to program the processor with only one mouse click directly from the simulation. This way the implementation cycle times for different types of algorithms are reduced dramatically. As different algorithms can run in different sample times a real-time Linux with the ability of multitasking has been considered. Following a closer look at the hardware the main points of the real-time Linux are briefly explained. The execution times of the Kalman navigation filter and the spline controller are elaborated and the necessity of multitasking is explained in this scope. Starting with a simulation of the airplane with navigation and control the complete simulation can be tested as software in the loop on the autopilot hardware.

A result of the symbiosis of all components is that complex algorithms from a high-level GUI environment can be elaborated in flight tests, as the programming from the object oriented high-level GUI interface to the ready to flight UAV takes around 30 seconds. At the end of this paper, results of the navigation solution and control algorithms are shown in experimental flight test.

http://www.imavs.org/

2 The Autopilot System

The autopilot hardware (see fig. 2) is based on a two processor concept. As the main computing processor an OMAP3530 [8] is used, which offers a good computing performance with 720 Mhz and a floating point unit. This main computing processor is connected to the data acquisition processor STM32 [14] with 72 Mhz and the capability of hardware interfaces like the CAN-Bus.

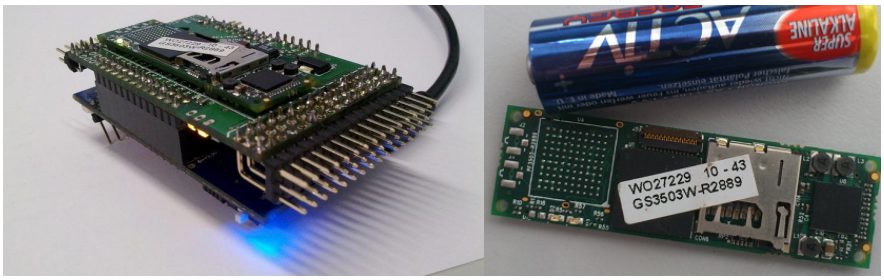


Figure 2: Left: Autopilot Hardware, Right: OMAP3530 mounted on a gumstix board

As the STM32 uses no operating system, its software uses only one thread. Hardware interrupts can be programmed directly beneath the main processor loops and no high level driver need to be used as the hardware can be programmed directly through the processor registers. Thus the STM32 software setup is relative simple to the OMAP3530 which uses Linux as operating system with multithreading for complex algorithms and high level drivers for the peripheral interfaces. In the case of an OMAP3530 failure, e.g. an experimental algorithm crashes the system, the STM32 will always be able to take control over the airplane. As all actuators and sensors are connected to the STM32 (see fig. 3) and as a high update rate for algorithms is preferable, also a high data bandwidth between the STM32 and the OMAP3530 is of benefit. For this reason an SPI (serial peripheral interface) Bus has been chosen which

allows bidirectional 20 MBit/s which is sufficient for the transmission of 50 signals for sensors and actuators in both directions within 300 μs.

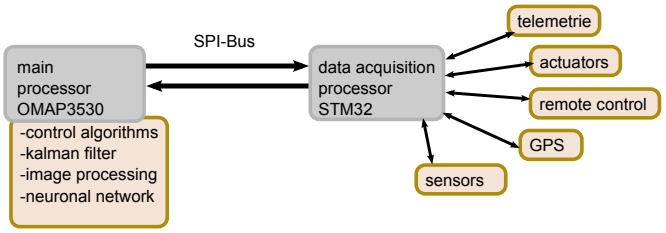


Figure 3: Connection scheme between processors

2.1 Operating System

The OMAP 3530 processor is running a real-time Linux [1] [13] which allows the usage of multithreading and complex drivers like a TCP stack or WiFi drivers. The fastest thread, i.e. the navigation and control algorithms, are running with an update rate of 100 Hz. This means that this thread must be executed deterministically within 10 ms, which leads to the most important variable of the real-time Linux, the execution delay: A standard Linux kernel behaves non deterministically [2] and schedules the given tasks, respectively the threads in a fair manner. Although the tasks are executed as fast as possible, it is not predictable when or how long they are executed. To overcome these problems a real-time layer has been integrated by the kernel developers which gives the ability to run high priority tasks with a deterministic execution behavior. Figure 4 shows the bounded the thread execution without real-time priority that can be very high whereas with real-time the delay is bounded below $< 65\mu$.

The STM32 runs its own independent 100 Hz loop. If the OMAP3530 provides a SPI-connection, the STM32 will transfer the sensor data to the OMAP3530 and use the computed control commands for the actuators (compare 4). The OMAP3530 runs several threads for different frequencies, e.g. the Kalman filter for navigation with 100 Hz. Non real-time threads can be executed in parallel e.g. to run a terminal via WiFi.

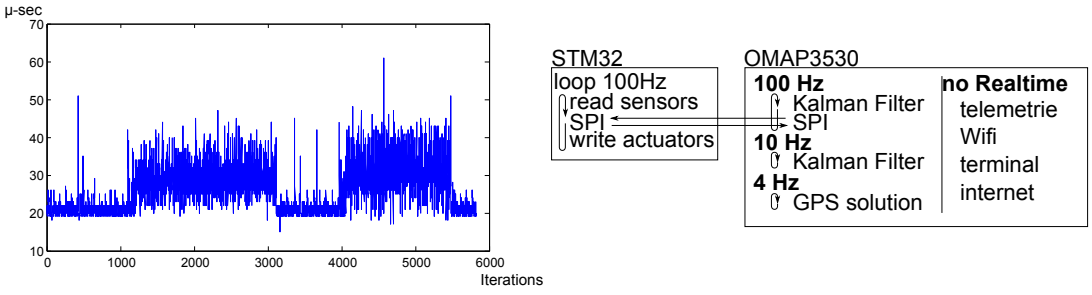


Figure 4: Left: The execution delay is bounded below $< 65\mu$, Right: Different threads running on the two processors

http://www.imavs.org/

http://www.imavs.org/

2.2 From Simulation to Flight Experiment

Coming from the hardware and software description, the concept of the autocode [12] design is elaborated. The algorithms for navigation and control are designed in the object oriented simulation environment Matlab/Simulink. Figure 5 depicts a simple simulation where a sinus signal is added to a constant matrix and the result is plotted in a "scope". To run this simulation on the autopilot system a so called tool-chain is used, where the simulation is translated into C-Code and compiled into machine executable binaries. To this purpose for example an "add-block" (compare fig. 5) is translated by Matlab's Target Language Compiler (TLC) with its corresponding tlc file into a C-Code function. In this way the simulation with all its blocks

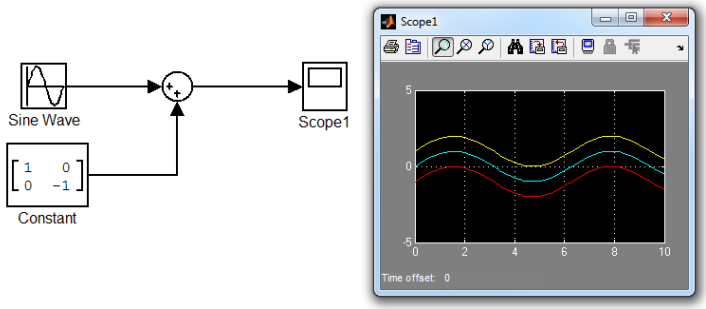


Figure 5: A simple Simulink model

creates a C-Code program, e.g. airplane.c (compare fig. 6). In a last step the compiler tool-chain for the specific processor translates the C-Code into a machine readable binary. Hardware like actuators or sensors can be modeled as a simulink block and after programming the autopilot this hardware is directly used. With the so called "external mode" it is possible to link and monitor the simulation on the real hardware via WiFi in the "scopes" of the developement hardware.

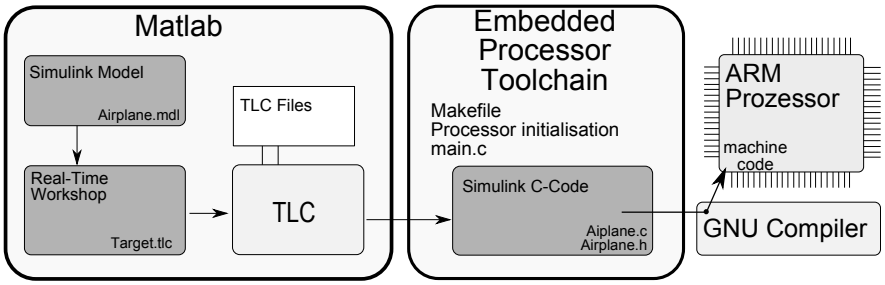


Figure 6: Autocode Environment: from model to machine code

2.3 Simulation

Before going into the flight experiment the navigation and control algorithms are tested in the simulation. The simulation is built out of five main parts (fig: 7): airplane, sensors, navigation,

control and actuators. The main part is the airplane, where actuator signals and wind simulation influence the airplane's forces and moments, which are integrated twice into position and attitude [3]. The airplane variables (position (4 Hz), rotational velocities, accelerations) are measured by the sensors. This sensor simulation includes sensor noise, temperature drifts or position errors. With this data the navigation reconstructs an attitude signal and a 100 Hz position for the controller cascade. The controller compares the actual position with the desired flight path and generates control signals for the actuators. The actuators are usually servos or the engine, which includes delays and nonlinearities from the input signal to the outgoing force. The connection to the airplane closes the loop.

In the development process the simulation is tested on the development environment: Running the simulation here takes less than a second and the simulation is running faster than real-time. If the algorithms perform good in the simulation, they are compiled and programmed onto the autopilot and tested in real-time, which is usually called software in the loop.

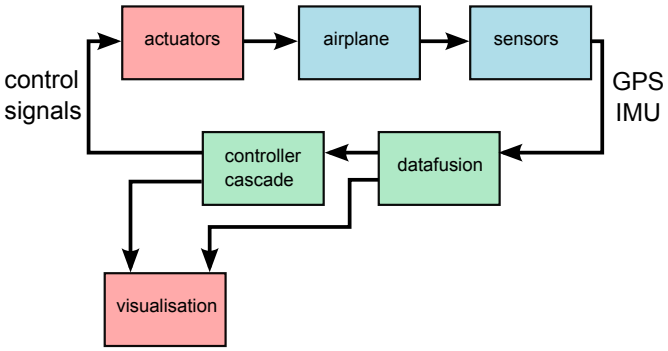


Figure 7: Airplane simulation with navigation and control

2.4 Kalman Filter

To control the airplane a precise position and attitude signal is necessary. As the dimensions of the small UAV require small and lightweight hardware, Microelectromechanical (MEMS) sensors are used in the Inertial Measuring Unit (IMU). The used low-cost acceleration and gyro sensors operate with a frequency of 100 Hz but suffer from drifting signals.
To compensate this drifting, long-term stable 4 Hz position measurements of the the GPS receiver are used. In this way position, velocity and attitude can be determined at the rate of the IMU with a better precision than with a stand-alone GPS receiver (fig: 8).

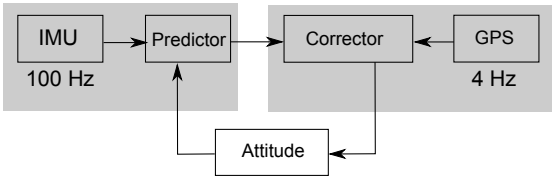


Figure 8: Kalman Filter

Different concepts for the fusion of sensor data as from GPS and IMU measurements exist

http://www.imavs.org/

(see [6]). The simplest and therefore most common integration is the so called loosely coupled system. These systems use the position and velocity information of the GPS to bind the IMU measurement errors [5].

The Kalman filter, introduced in [10], has become a quasi-standard for accomplishing the data fusion of inertial and satellite navigation. The navigation system of the new autopilot system is based on a time discrete, loosely coupled extended Kalman filter, see [4]. The error state architecture allows the estimation of a non-linear process with a linear Kalman filter.

The utilized state vector $\vec{x}$ consists of 15 states for errors in position, velocity, attitude, gyro bias, and accelerometer bias.

$$\vec{x} = \begin{bmatrix} \delta r_g^{eb} & \text{... 3 position errors} \\ \delta v_g^{eb} & \text{... 3 velocity errors} \\ \delta \phi^{gb} & \text{... 3 attitude angle errors} \\ \delta \omega_b & \text{... 3 errors in gyro bias} \\ \delta a_b & \text{... 3 errors in accelerometer bias} \end{bmatrix}$$

The indices g and b show that the vector is described in the geodetic- and the body-frame respectively. The superscripts and stand for the body- with respect to earth-centred earth-fixed (index e) (ECEF)-frame and b -with respect to -frame, respectively.

The Kalman filter works in two phases - propagation and estimation, which justifies the need of multithreading. The propagation is executed at the IMU's measurement frequency of 100 Hz. Parallel to the propagation process, the navigation solution is calculated using the IMU measurements. These are processed via a so-called strapdown algorithm, which allows the computation of navigation data from body-fixed inertial sensors, see [9].

The estimation process is started when new GPS measurements have arrived. During this update, the measurements $\vec{z}_k$ which are received at time t_k are processed. This vector consists of the differences between predicted and measured values of the position.

A hat $\hat{}$ indicates values which are re-estimated, measured values are denoted by a tilde $\tilde{}$ on top of each variable. The enhancement of the navigation data quality is performed by the calculation of the so called Kalman Gain matrix and the following update of the state vector and the covariance matrix of the state estimation uncertainty.

$$K_k = P_k^{-1} \cdot H_k^T \cdot (H_k \cdot P_K^{-1} \cdot H_k^T + R_k)^{-1}$$

$$\hat{\vec{x}}_k^+ = \vec{x}_k^- + K_k \cdot (\tilde{\vec{z}}_k - H_k \cdot \vec{x}_k^-)$$

$$P_k^+ = (I - K_k \cdot H_k) \cdot P_k^-$$

The Kalman Gain matrix weights the difference between measurements $\vec{z}_k$ and the expected measurement, which is calculated by the a priori state vector and the measuremnt matrix $\underline{H}_R$. For this it includes the autocovariance matrix of the measurement noise $\underline{R}_k$. The measurement matrix maps the state vector onto the measurement vector. Matrix $\underline{I}$ is a 15 $\times$ 15 identity matrix.

http://www.imavs.org/

2.4.1 Sensor Calibration

Some attention needs to be given to the sensor calibration. The actual autopilot uses an IMU3000 Gyro from Invensense and an Analog Devices ADXL345 acceleration sensor. Those sensors deliver an unsigned binary number proportional to the measured variable. The sensor output $w_{i,meas}$ of the gyroscopes has been modeled as follows:

$$w_{i,meas} = w_i * a_{w,i} + b_{w,i} + c \cdot T \tag{1}$$

where w_i denotes the angular rate for an axis i , which is multiplied by a scaling factor $a_{w,i}$. An static bias $b_{w,i}$ is added to a temperature depended bias $c \cdot T$. The used accelerometers have been modeled without the temperature dependent bias with the static bias $b_{a,i}$ and the scaling $a_{a,i}$ to the equation:

$$a_{i,meas} = a_i * a + b. \tag{2}$$

Simulation shows that this sensor model is sufficient for navigation and control of the UAV (compare fig. 9). The sensor calibration followed a simple scheme:The autopilot is rotated on

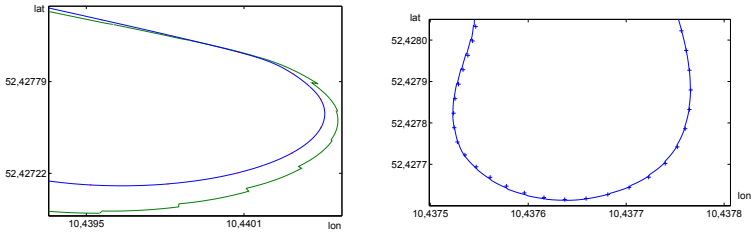


Figure 9: Left: navigation simulation with no sensor errors (blue), navigation with 30% error in bias value (green); Right: navigation in real flight (line), GPS measurements (dots)

the Desk around every axis with 90°. In a first step the gyro signal is out-biased, which means, the end value of a forward and back rotation on the Desk is known as zero. The signal is scaled to fit 90°, compare figure 10. As there a barely visible temperature effect in the scaling parameter, only the gyro bias is considered. This was done by putting the autopilot into a fridge while the bias drift was recorded.

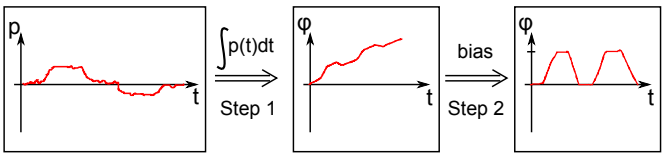


Figure 10: Step 1: Summarize gyro rate signal into angle, Step 2: Calculate bias, Step 3: Scaling

2.5 Spline Control

To be able to follow a predefined path a spline controller is used. To this purpose two-dimensional cubic bezier splines are used. Thus the curves follow a polynome of third degree [7].

http://www.imavs.org/

The spline can be described with the Point $A_n = [x_n, y_n]$ as follows:

$$x(t) = a_3t^3 + a_2t^2 + a_1^t + x_0 \tag{3}$$

$$y(t) = b_3t^3 + b_2t^2 + b_1^t + y_0 \tag{4}$$

The run parameter t lies in the region between $0 \leq t \leq 1$. The spline parameters are defined as follows:

$$a_1 = 3(x_1 - x_0), a_2 = 3(x_0 - 2x_1 + x_2), a_3 = (-x_0 + 3x_1 - 3x_2 + x_3) \tag{5}$$

$$b_1 = 3(y_1 - y_0), b_2 = 3(y_0 - 2y_1 + y_2), b_3 = (-y_0 + 3y_1 - 3y_2 + y_3) \tag{6}$$

The spline path is preset in the ground-station software (see fig. 11) and transmitted via WiFi or radio to the UAV.

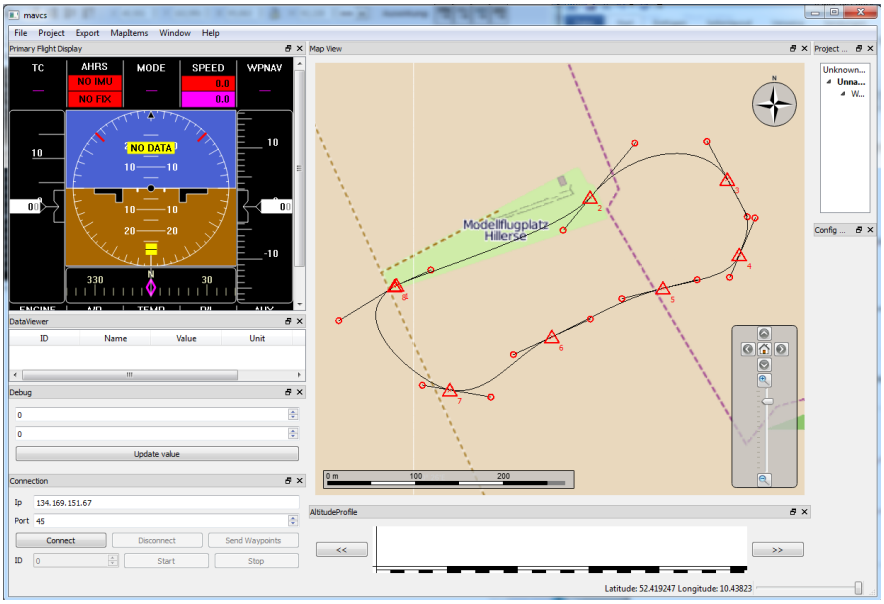


Figure 11: Ground-station GUI with flight display and waypoint navigation map

2.6 Flight experiments

As experimental setup a Multiplex Twinstar airplane model (see fig.1 right) was used. This model has a wingspan of 1.4 meters, and a take off weight of 1.5 kg. The airplane speed ranges from 35km/h to 70km/h. The experiments were made on a moderate gusty day with a wind speed of 15km/h.

2.6.1 Attitude Controller Flight Result

Usually the airplane model operator controls the angular rates of the model with the remote control and thus the elevator and aileron position. In the attitude controller mode the remote

http://www.imavs.org/

control stick positions for the aileron u_{aile} and the elevator u_{ele} are used as reference signals for the roll Φ and pitch Θ angle of the airplane model, such that the aileron and the elevator position $(\varphi_{aile}, \varphi_{ele})$ results directly from this difference:

$$\varphi_{ele} = \Phi - u_{ele} \tag{7}$$

$$\varphi_{aile} = \theta - u_{aile} \tag{8}$$

In the Attitude controller mode taking off and landing the airplane is possible even for novice pilots. Figure 12 plots the aileron signal u_{aile} and the measured roll angle Φ. As the angle is used directly as control feedback, no damping terms are considered which leads to visible overshoots in the angle.

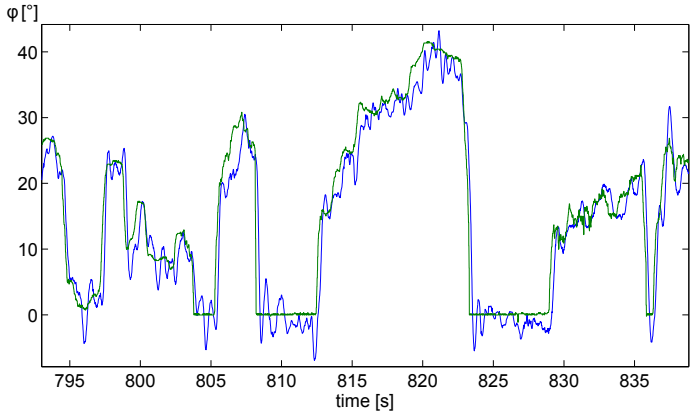


Figure 12: Attitude controller: roll reference signal u_{aile} (green), roll Φ estimated from navigation (blue)

2.6.2 Spline controller flight result

After taking off the airplane in the attitude controller mode, the spline controller is turned on. In this mode the UAV is flying automatically the predefined path. The navigation and control algorithms consume $\sim$ 25% of the processor time on the OMAP3530. Terminal via WiFi or UMTS is always accessible and the flight data is recorded in a non real-time thread. Figure 13 depicts the flight path of the UAV in this experiment. Usually the spline controller is started after takeoff in some distance to the pre-programmed flight path. If the distance to this flight-path (respectively spline-path) is high, the spline controller cascade is going into saturation such that an additional start spline is calcualated automatically in the autopilot from the actual UAV position to the beginning of the predefined flight path.

2.7 Conclusion and Outlook

The combination of Linux and auto-code systems leads to a very effective experimental platform. The development cycles from simulation to the flight experiment are reduced to less than a few minutes. A spline controller in combination with a Kalman filter is running easily on the modern 720 Mhz cell phone processor OMAP3530 and leaves enough resources for e.g. running neural networks. It is expected that actual cell phones with multicore processors and a real-time

http://www.imavs.org/

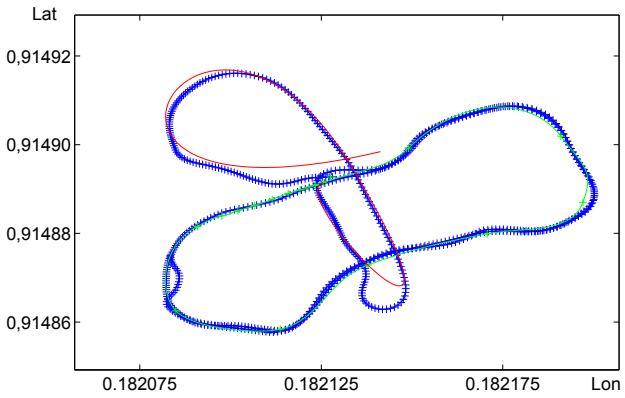


Figure 13: Flight experiment: flight path with GPS measurements (blue line and blue marks), spline (green line), transition spline (red line)

Linux kernel have the capabilities to control UAV's with enough resources for more complex and intelligent algorithms.

References

[1] Stefan Agner. *Linux Realtime-Faehigkeiten.* Hochschule Luzern, *http : //www.agner.ch/linuxrealtime/Linux − Realtime − Faehigkeiten.pdf*, 2009.

[2] Cesati M. Bovet P. *Understanding the Linux Kernel.* Oreilly, *http : //oreilly.com/catalog/linuxkernel/chapter/ch*10.*html*, 2000.

[3] W. Alles Brockhaus, R. *Flugregelung.* Springer Verlag, Berlin, 2011.

[4] J. Crassidis. *Sigma Point Filtering for Integrated GPS and Inertial Navigation.* University at Buffalo, New York, U.S.A, -.

[5] J. Weston D. Titterton. *Strapdown Inertial Navigation Technology.* The Institution of Engeneering and Technology, London, 2004.

[6] J. L. Farrell and Barth. *The Global Positioning System and Inertial Navigation.* McGraw-Hill, New York, U.S.A, 1999.

[7] Hans-Georg-Buesing. *Ein Flugregler fuer die Startphase autonomer Kleinstflugzeuge.* TU Braunschweig, Braunschweig, 2006.

[8] Texas Instruments. *OMAP 3530 Application processor.* Texas Instruments, Silicon Valley, 2013.

[9] Fried Kayton, M. *Avionics Navigation Systems.* John Wiley and Sons, New York, U.S.A., 1997.

[10] R. E. Klmn. *A New Approach to Linear Filtering and Prediction Problems.* Journal of Basic Engineering, Baltimore, U.S.A, 1960.

[11] A. Kroonberg. *Airborne Measurement of Small-Scale Turbulence with special regard to the Polar Boundary Layer.* Zentrum fr Luft- und Raumfahrt, Braunschweig, 2009.

[12] Mathworks. *Embedded Coder User Guide.* Mathworks, *http : //www.weizmann.ac.il/matlab/pdf*$_d$*oc/ecoder/ecoder*$_u$*g.pdf*, 2013.

[13] OSADL. *Real-Time Linux Wiki.* Open Source, *https : //rt.wiki.kernel.org/index.php/Main*$_P$*age*, 2013.

[14] STMicroelectronics. *STM32 Mainstream Performance line, ARM Cortex-M3 MCU.* STMicroelectronics, Silicon Valley, 2013.

http://www.imavs.org/

Paparazzi: how to make a swarm of Parrot AR Drones fly autonomously based on GPS.

Bart Remes, Dino Hensen, Freek van Tienen, Christophe De Wagter,
Erik van der Horst, and Guido de Croon

Micro Air Vehicle laboratory, Delft University of Technology, Kluyverweg 1, 2629 HS, Delft, the Netherlands.
Contact: microuav@gmail.com

http://www.imavs.org/

Abstract

The Parrot AR drone 2 is a commercially available quad rotor with many sensors onboard, such as two cameras, accelerometers, gyrometers, barometer, and a magnetometer. Recently, a GPS-module has become available, which can be used to geo-tag images. We have adapted the particularly versatile and powerful Paparazzi open source autopilot for use with the Parrot AR drone. Hence, just by uploading software to the drone, it is able to perform GPS-based autonomous outdoor flight. In this article, we explain the inner workings of the adapted autopilot and what procedure a user should follow for performing experiments with a swarm of AR drones.

1 Introduction

In the past decade, the fast developments in micro-electronics and the related price reductions have led to Micro Air Vehicles (MAVs) able to perform GPS-based outdoor flight. The low cost of the necessary components and the creation of open source autopilot projects give a growing group of hobbyists and researchers the opportunity to construct and fly their own MAVs [8]. Example projects include ArduPilot [1], OpenPilot [2] and Paparazzi (cf. [3, 6, 13]).

In the same time, toy planes and helicopters have become more and more sophisticated. As an ultimate example, one can think of the Parrot AR drone [4, 7], which is equipped with two cameras, accelerometers, gyros, sonar, and sufficient onboard processing to autonomously cancel drift if there is (visual) texture on the ground surface. The availability of these sensors, the relatively low price, and an open source Standard Development Kit (SDK) have resulted in the robotics community making extensive use of the Parrot AR drone for their experiments (cf. [5, 10, 14, 9]). However, these experiments have been limited to flights without GPS-based navigation.

In order to facilitate MAV research with GPS-enabled flight, we have recently adapted the particularly versatile and powerful Paparazzi autopilot for use with the Parrot AR drone. The autopilot works completely with the Parrot hardware, i.e., the drone itself and the GPS-module. Hence, no hardware modification is necessary[1].

In this technical paper, we first explain the straightforward procedure for enabling GPS-based flight with the Parrot AR drone (Section 2). Subsequently, we provide insight into the basic inner-loop and outer-loop control of the now GPS-enabled drone (Section 3), and explain how to set up an experiment with a swarm of drones (Section 4). We conclude in Section 5.

[1]Please note that the explained software project is not supported nor endorsed by Parrot S.A. Using the software will most likely void the warranty and use of the software is entirely at the user's own risk.

2 How to make the Parrot AR drone fly with GPS

There are two ways to make the AR drone fly with GPS: (1) enhancing the Parrot firmware with Paparazzi firmware (the *Paparazzi AR drone SDK* version), (2) only using Paparazzi firmware (the *Paparazzi AR drone raw* version). The difference between these two versions is explained in more detail in Section 3. For now, it is important to know that the raw version is also called the expert version, requiring some more work but also providing more control of what is happening onboard the AR drone. Below, we explain for both versions how Paparazzi can be installed. Please remark that the Paparazzi wiki has a more elaborate, up-to-date explanation of these procedures [2,3].

2.1 Paparazzi AR drone SDK version

The only requirements besides an AR Drone 2, are a Parrot GPS receiver and a laptop with Ubuntu Linux OS installed (a virtual machine can be used to this end). To be fully compatible you need at least ARDrone 2 software v2.4.3 updated on your AR Drone 2 and it is best to have Linux v12.04 LTS on your laptop.

The steps to enable GPS-based flight are:

1. Install Paparazzi. Please remark that you need to check out the master version, and that you need to allow access for USB flashing with a few commands explained on the wiki[4].
2. Launch Paparazzi by clicking on the desktop item.
3. Insert the Parrot GPS module into your ARDrone2 and power it up by connecting the battery.
4. Establish a Wifi connection between the ARDrone 2 and your laptop. Select ardrone2 v2.4.3 in Wireless Network List.
5. At Paparazzi Center select ardrone2 sdk at the "A/C" drop-down box. Make sure "ap" (Autopilot) is set as Target.
6. Press the "Upload" button.
7. In the Paparazzi control center select "ARDrone 2 Flight" as session.
8. Press "Execute"..

After the above-mentioned steps, the Parrot AR drone 2 is ready for GPS-enabled autonomous flight. It is the responsability of the operator to ensure the safety of autonomous flight in accordance with local regulations. It is important to realize that GPS-measurements can have uncertainties of several meters, so it is best to test in areas with sufficient available space.

When going outside after performing the above-mentioned steps, the autopilot will attempt to lock on to sufficient GPS-satellites. After waiting for about a minute a GPS signal will be found. The ARDrone2 location then appears in the map of the Ground Control Station (GCS). When flying for the first time in a certain location, one needs to download the Google map tiles on to the GCS computer. This can be done in advance. Figure 1 shows a screenshot of the Paparazzi GCS for autonomous flight with the AR drone 2. The flight can start by pressing "Takeoff". The remainder of the flight will happen according to the Paparazzi flight plan that the user has constructed for the mission. More information on flight plans can be found on the Paparazzi wiki[5].

[2] http://paparazzi.enac.fr/wiki/AR_Drone_2/getting_started
[3] http://paparazzi.enac.fr/wiki/AR_Drone_2/getting_started_advanced
[4] http://paparazzi.enac.fr/wiki/AR_drone_2/Software_installation
[5] http://paparazzi.enac.fr/wiki/Main_Page

http://www.imavs.org/



Figure 1: View of the Ground Control Station of Paparazzi for the Parrot AR drone 2. The overlays include current information of the drone's state such as its height, controls such as the home-button, and the waypoints in the flight plan.

2.2 Paparazzi AR drone raw version

The requirements for using the Paparazzi AR drone raw version has in principle the same requirements as the Paparazzi AR drone SDK version. However, best results can be obtained if a different GPS receiver is used, such as a u-Blox with high update rate GPS. In addition, it is recommendable to have a a joystick.

The steps to install the raw version are then, in brief:

1. Install Paparazzi. In this case, for now one has to download a specific, TU Delft "fork" of the code.
2. Install the joystick.
3. Connect the joystick with your computer.
4. Connect the GPS with the AR.Drone 2 USB port.
5. At paparazzi center select at "A/C" the ardrone2.
6. Select "AP" as Target and press Build.
7. Turn the AR.Drone 2 on by connecting the battery.
8. Connect your computer with the AR.Drone 2 trough WiFi.
9. Select "Flight Wifi" as session and press the Execute button.

http://www.imavs.org/

10. At paparazzi center press the "Upload" button. This step will automatically 'kill' the Parrot firmware running onboard the drone and install Paparazzi flight software and drivers. Please note that upon restart, the Parrot AR drone firmware is restored automatically. So the Parrot firmware is not lost.

11. Calibrate the compass by having the drone perform a full turn. The details of this procedure are explained on the Paparazzi wiki.

3 Inner-loop and outer-loop control of GPS-enabled AR drones

In this section, we give some insight into the control loops involved in both Paparazzi versions for the AR drone. In the Paparazzi AR drone SDK version, only the outer loop control is performed by Paparazzi. This implies that Paparazzi ensures the execution of the flight plan and makes the drone move from waypoint to waypoint, but that all lower-level control remains in the hands of the Parrot firmware. For example, the sensor fusion is performed by the Parrot firmware: the barometer, sonar, accelerometers, gyrometers, and magnetometer are integrated with optic flow estimations from the bottom camera. Combined with the barometer and sonar, the optic flow estimations provide accurate estimates of the drone's speeds. Paparazzi regulates the vertical speed, attitude angles, and yaw rate of the drone via 'AT-commands' from the Parrot SDK. Currently, Paparazzi's rotorcraft control sets the velocity of the drone in the direction of the next waypoint, which implies that the drone will roll and pitch to fly in the right direction. Parrot's firmware receives the AT-commands and controls the velocities and attitude. The use of Parrot's firmware for inner loop control provides a user the ability of GPS-enabled flight, but limits the possibilities to change and analyze this level of control. Namely, the Parrot firmware is closed source.

In the Paparazzi AR drone raw version, both the outer loop and inner loop control is performed by Paparazzi, providing the user with more possiblities especially useful for research. The version currently available on the Paparazzi repository can read out and use all sensors via Parrot's sensor board, ranging from the accelerometers to the barometer, except for the camera's. Moreover, the current rotorcraft state estimation does not use the sonar by default. Figure 2 shows a global overview of the control loops involved. The safety pilot has control over the vehicle via the RC link, which in the case of the Parrot AR drone means the joystick connected to the GCS. The flight plan controls the thrust and attitude for achieving the right speed and position. All control loops are open source and can be changed by the user[6]. The standard Paparazzi version makes extensive use of well-known PID-loops, but git-branches are available that implement more complex forms of control.

4 How to search for a target with a swarm of AR drones

Many research groups perform research on *swarming* algorithms (cf. [12, 11, 15]). The developed Paparazzi versions for the AR drone now provide the opportunity to research groups to test these algorithms on real platforms. Here we explain how to connect multiple AR drones to one GCS.

The following steps then suffice to perform a swarming experiment:

1. Add this line to the AR Drone 2 Airframe file:
   <define name="LINK_HOST" value="\\\"192.168.2.0\\\"" />

[6] `http://paparazzi.enac.fr/wiki/Control_Loops`

http://www.imavs.org/

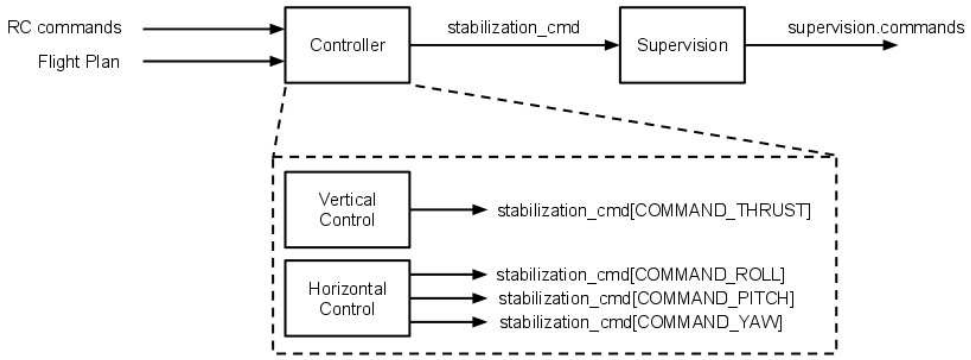


Figure 2: Global overview of the control loops involved in the Paparazzi AR drone raw version (see the text for a detailed explanation).

2. Switch on the first AR Drone
3. Connect to the Drone via Wifi SSID with name "ardrone2_XYZ" where XYZ is the current SSID of the drone.
4. Upload the Paparazzi autopilot into the AR drone as explained in previous sections.
5. Replace XYZ with the wished IP number of your ARDrone, e.g. 123 in the line below. Give the following command in the Linux terminal: *echo "killall udhcpd && ifconfig ath0 down && iwconfig ath0 mode managed essid hier ap any channel auto && ifconfig ath0 192.168.2.XYZ netmask 255.255.255.0 up && sleep 1" — telnet 192.168.1.1*
6. After this, connect the laptop Wifi to the router via SSID "thenameoftheaccespoint"
7. If all was done correctly, one of the AR drones is now connected to the swarm access point.
8. To test, type this in your terminal: *ping 192.168.2.XYZ*
9. To add more MAVs to the swarm, just repeat the steps above for every new AR drone 2 where one assigns a new IP for every new drone.

Figure 3 shows the GCS when there are five drones connected to the router. Now the flight plan can be adjusted to performing swarming experiments with the Parrot AR drone 2!

5 Conclusion

We have presented and explained two different Paparazzi versions that allow the Parrot AR drone 2 to fly autonomously outdoors on the basis of GPS. The Paparazzi AR drone SDK version performs the outer loop control for executing a flight plan, but makes use of Parrot's firmware for sensor fusion and inner-loop control. Although these functions are well-implemented in the firmware, the code is

http://www.imavs.org/

http://www.imavs.org/

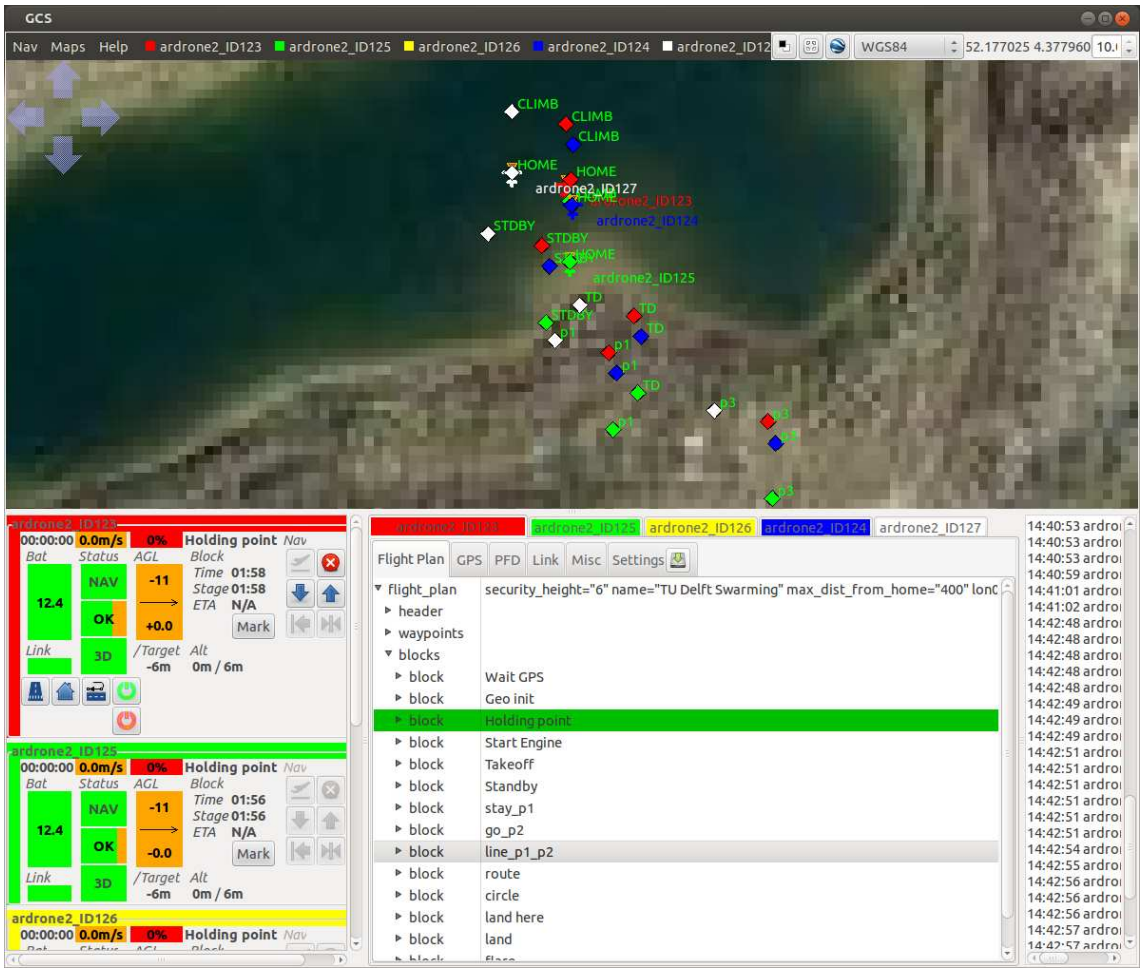


Figure 3: GCS with five AR drones.

closed source and cannot be adapted for research purposes. The Paparazzi AR drone raw version performs both the outer loop and the inner loop control. The currently available version has access to all sensors, except for the cameras. Both versions allow the connection of multiple drones to the GCS via a router. This means that research groups all over the world can now perform swarming research with a group of relatively affordable MAVs.

Finally, we want to remark that we have just gained access to the cameras onboard the AR drone. This implies that researchers can also perform onboard vision processing on the Parrot AR drone, by adding vision algorithm code to the drone's firmware. We are now working on an easy integration of this code into Paparazzi. Accessing the cameras and adding vision algorithms will be discussed in a follow-up paper.

References

[1] http://diydrones.com/.

[2] http://www.openpilot.org/.

[3] http://paparazzi.enac.fr/wiki/.

[4] http://ardrone2.parrot.com/.

[5] C. Bills, J. Chen, and A. Saxena. Autonomous MAV flight in indoor environments using single image perspective cues. In *ICRA*, pages 5776–5783, 2011.

[6] P. Brisset, A. Drouin, M. Gorraz, P.S. Huard, and J. Tyler. The paparazzi solution. In *MAV2006*, 2006.

[7] P.J. Bristeau, F. Callou, D. Vissière, and N. Petit. The navigation and control technology inside the ar. drone micro UAV. In *18th IFAC World Congress*, pages 1477–1484, 2011.

[8] H. Chao, Y. Cao, and Y. Chen. Autopilots for small unmanned aerial vehicles: a survey. *International Journal of Control, Automation and Systems*, 8(1):36–44, 2010.

[9] G.C.H.E. de Croon and S. Nolfi. Act-corner: Active corner finding for optic flow determination. In *ICRA*, 2013.

[10] N. Dijkshoorn and A. Visser. Integrating sensor and motion models to localize an autonomous ar drone. *International Journal of Micro Air Vehicles*, 3:183–200, 2011.

[11] A. Kushleyev, V. Kumar, and D. Mellinger. Towards a swarm of agile micro quadrotors. In *Robotics: Science and Systems*, 2012.

[12] L.M. Gambardella F. Mondada S. Nolfi T. Baaboura M. Birattari M. Bonani M. Brambilla A. Brutschy D. Burnier A. Campo A.L. Christensen A. Decugnire G.A. Di Caro F. Ducatelle E. Ferrante A. Frster J. Guzzi V. Longchamp S. Magnenat J. Martinez Gonzalez N. Mathews M.A. Montes de Oca R. O'Grady C. Pinciroli G. Pini P. Rtornaz J. Roberts V. Sperati T. Stirling A. Stranieri T. Stuetzle V. Trianni E. Tuci A.E. Turgut F. Vaussard M. Dorigo, D. Floreano. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics and Automation Magazine*, 2012.

[13] J. Reuder, P. Brisset, M. Jonassen, M. Muller, and S. Mayer. The small unmanned meteorological observer sumo: A new tool for atmospheric boundary layer research. *Meteorologische Zeitschrift*, 18(2):141–147, 2009.

[14] S. Ross, N. Melik-Barkhudarov, K.S. Shankar, A. Wendel, D. Dey, J.A. Bagnell, and M. Hebert. Learning monocular reactive uav control in cluttered natural environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[15] J. Zufferey D. Floreano T. Stirling, J. Roberts. Indoor navigation with a swarm of flying robots. In IEEE, editor, *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, 2012.

http://www.imavs.org/