Incremental Control of Hybrid Micro Air Vehicles

Smeur, Ewoud

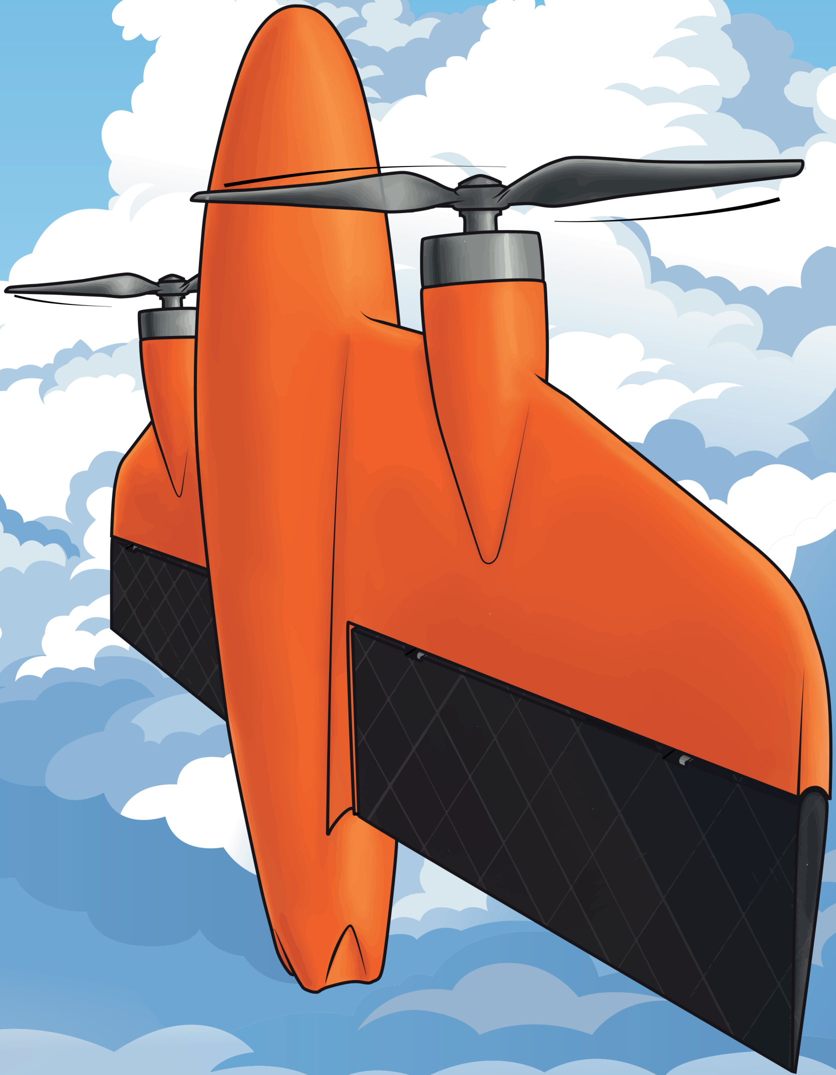
DOI
10.4233/uuid:23c338a1-8b34-40a6-89e9-997adbdafd75

Publication date
2018

Document Version
Final published version

Citation (APA)
Smeur, E. (2018). *Incremental Control of Hybrid Micro Air Vehicles*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:23c338a1-8b34-40a6-89e9-997adbdafd75

Incremental Control for Hybrid Micro Air Vehicles



Ewoud Jan Jacob Smeur

Incremental Control of Hybrid Micro Air Vehicles

Incremental Control of Hybrid Micro Air Vehicles

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 5 november 2018 om 15:00 uur

door

Ewoud Jan Jacob SMEUR

Ingenieur Luchtvaart en Ruimtevaart,
Technische Universiteit Delft, Delft, Nederland,
geboren te Nijmegen, Nederland.

Dit proefschrift is goedgekeurd door de

promotor: prof. dr. ir. J.M. Hoekstra
copromotor: dr. G.C.H.E. de Croon

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. J.M. Hoekstra,	Technische Universiteit Delft, promotor
Dr. G.C.H.E. de Croon,	Technische Universiteit Delft, copromotor
Onafhankelijke leden:	
Prof. dr. ir. D. Alazard	ISAE, France
Prof. dr. ing. F. Holzapfel	TU München
Prof. dr. ing. D. Moormann	Tech. Hochschule Aachen, Germany
Prof. dr. R. Babuska	Technische Universiteit Delft
Prof. dr. ir. M. Mulder,	Technische Universiteit Delft, reservelid
Overige leden:	
Dr. Q.P. Chu,	Technische Universiteit Delft

TU Delft Delft University of Technology

Delphi Consortium

Keywords: Incremental nonlinear dynamic inversion, hybrid micro air vehicles, tailsitter, control allocation

Printed by: Ipskamp

ISBN 978-94-6186-973-9

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Contents

Summary ix

Samenvatting xi

1 Introduction 1

1.1 Types of hybrid MAVs 2
1.2 Tailsitter hybrids 4
1.3 Control of hybrids 5
1.3.1 Attitude control 5
1.3.2 Velocity control 6
1.4 Incremental Nonlinear Dynamic Inversion 6
1.5 Outline 8

2 Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles **11**

2.1 Introduction 12
2.2 MAV Model 14
2.3 Incremental Nonlinear Dynamic Inversion 16
2.3.1 Parameter Estimation 18
2.3.2 Implementation 19
2.3.3 Closed Loop Analysis 20
2.3.4 Attitude Control 21
2.3.5 Altitude Control. 22
2.4 Adaptive INDI 22
2.5 Experimental Setup 23
2.5.1 Performance 24
2.5.2 Disturbance Rejection 24
2.5.3 Adaptation 25
2.5.4 Yaw Control 26
2.6 Results 26
2.6.1 Performance 27
2.6.2 Disturbance Rejection 28
2.6.3 Adaptation 30
2.6.4 Yaw control 33
2.7 Conclusion 35

3 Prioritized Control Allocation for Quadrotors Subject to Saturation **37**

3.1 Introduction 38

3.2 Incremental Nonlinear Dynamic Inversion and Actuator Saturation . . . 39
3.3 Using the Active Set Method to Solve the Constrained Allocation Problem . . . 40
3.3.1 Particularities for WLS applied to INDI . . . 43
3.3.2 Choice of Weighting Matrices . . . 43
3.3.3 Computational Complexity . . . 44
3.4 Experiments . . . 44
3.4.1 Experimental Setup . . . 44
3.4.2 Results . . . 45
3.5 Conclusion . . . 47

4 Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection 49
4.1 Introduction . . . 50
4.1.1 Related work . . . 52
4.2 Incremental nonlinear dynamic inversion for attitude control . . . 52
4.2.1 Attitude control . . . 54
4.2.2 Determining the control effectiveness . . . 57
4.3 Incremental nonlinear dynamic inversion applied to linear accelerations . . . 57
4.4 Implementation . . . 59
4.4.1 Position control . . . 60
4.4.2 Adaptive control effectiveness . . . 60
4.4.3 Estimation of the thrust . . . 61
4.4.4 Linearization . . . 62
4.4.5 Filtering . . . 63
4.4.6 Accelerometer bias . . . 64
4.5 Wind tunnel experiment . . . 64
4.5.1 Test setup . . . 64
4.5.2 Results . . . 66
4.6 Outdoor takeoff with wind . . . 68
4.6.1 Test setup . . . 69
4.6.2 Results . . . 69
4.7 Nonlinear increment . . . 71
4.7.1 Test setup . . . 71
4.7.2 Results . . . 72
4.8 Conclusions . . . 73

5 Incremental control and guidance of hybrid aircraft applied to the Cyclone tailsitter UAV 77
5.1 Introduction . . . 78
5.2 Attitude Control . . . 80
5.2.1 Center of Gravity . . . 81
5.2.2 Incremental Nonlinear Dynamic Inversion . . . 82
5.2.3 Control Effectiveness Scheduling . . . 83

5.2.4 Effectiveness of thrust on pitch . . . 84
5.2.5 Effectiveness of propellers on rotation around Z axis . . 85
5.2.6 Control effectiveness matrix . . . 85
5.2.7 Control Allocation . . . 85
5.2.8 Knife-edge flight . . . 86
5.3 Sideslip. . . . 87
5.3.1 Estimating the sideslip angle . . . 88
5.3.2 Sideslip control . . . 89
5.4 Velocity Control. . . . 90
5.4.1 Effectiveness of the flaps on the lift . . . 94
5.4.2 Attitude gains in forward flight. . . . 97
5.5 Guidance . . . 98
5.5.1 Efficient turning . . . 98
5.5.2 Approaching a waypoint. . . . 99
5.5.3 Line following . . . 99
5.6 Test flight results . . . 100
5.6.1 Attitude control performance . . . 101
5.6.2 Forward flight . . . 102
5.7 Guidelines for implementing INDI for hybrids . . . 103
5.8 Conclusion . . . 105

6 Conclusion 109
6.1 Research Questions . . . 109
6.2 Discussion. . . . 111
6.3 Future Work. . . . 112

Acknowledgements 115
References. . . . 117

Curriculum Vitæ 125

List of Publications 127

Summary

Micro Air Vehicles (MAVs) can perform many useful tasks, such as mapping and delivery. For these tasks either rotorcraft are used, which can hover but are not very efficient, or fixed wing vehicles, which are efficient but can not hover. Hybrid MAVs combine the hovering of a rotorcraft with the efficiency of a fixed wing. The reason that these vehicles are not yet widely adopted is that they are very difficult to control.

This thesis addresses the use of Incremental Nonlinear Dynamic Inversion (INDI) for the control of the attitude and velocity of hybrid MAVs. This control method had not been applied in a real world application prior to this thesis, which is why the thesis encompasses the application to a quadrotor at first, which is easier to control than a hybrid MAV.

First, an INDI structure is proposed for the control of the angular accelerations of a quadrotor. I show that the delay that filtering of the angular acceleration produces should also be applied to the measurement of the actuator state. If this is done, the filtering does not appear in the transfer function from virtual control to angular acceleration, which turns out to be equal to the actuator dynamics. It is also shown that a disturbance, or unmodeled dynamics, is compensated with the transfer function of the actuator dynamics multiplied with the applied filter and a unit delay. Finally, it is shown how the effects of propeller inertia, which can be very significant in the yaw axis, can be dealt with and how the control effectiveness can be made adaptive. All these findings are validated with experiments on a Bebop quadrotor.

Second, this thesis includes a Weighted Least Squares (WLS) control allocation algorithm with priority management into the INDI controller. This means that for vehicles with coupled control effectors, certain control objectives can be given priority upon actuator saturation. This is very important for vehicles with controlled axes that are not very important for the stability of the vehicle, such as the yaw for a quadrotor. It is shown that for a quadrotor doing a 50 degree yaw change, the stability is greatly improved when the yaw axis is given very low priority.

Third, this thesis introduces the control of linear accelerations in all three axes with INDI. The controller does not need a complex model, but instead relies on a measurement of the acceleration. It is shown through a wind tunnel experiment, that the disturbance rejection properties, that were shown for the inner loop, carry over to the control of linear accelerations. It is also shown that the method can be applied outdoors with an off-the-shelve GPS receiver. Finally, a nonlinear method of calculating the input increment is derived, which provides only a slight improvement in the tracking of aggressive acceleration commands.

These three things are combined for the INDI control of hybrid MAVs. The result is a single, continuous INDI controller for the attitude, and a single, continuous

INDI controller for the velocity of the vehicle. This is achieved by incorporating partial derivatives of the lift vector in the control effectiveness of the pitch and roll angles. Though no transition maneuver is explicitly defined, the transition follows implicitly from the increments in attitude and thrust that are calculated from desired acceleration changes. Further, as the control effectiveness of a hybrid MAV changes dramatically over the flight envelope, the control effectiveness is scheduled as a function of airspeed. When the airspeed is too low to measure, the pitch angle is used for this purpose. To prevent sideslip, a sideslip controller is included, where an estimate of the sideslip angle is obtained from the accelerometer.

Test flights show that the INDI inner and outer loop controllers are indeed capable of controlling the attitude and the linear acceleration of the vehicle throughout the flight envelope, within the physical limitations of the vehicle. It is shown that the tracking of accelerations can make the vehicle naturally transition to forward flight and back, and fly in the stall region as necessary. Because of the abstraction that the INDI acceleration control provides, it is straightforward to follow a velocity vector field, for example one that guides the aircraft along a line.

The developed controller can be applied to different tailsitter MAVs with relative ease, as the model dependency is low. The algorithm may even be applied to different types of hybrid vehicles, such as quadplanes or tilt-wing aircraft, with minor adjustments.

Samenvatting

Microdrones kunnen veel nuttige taken vervullen, zoals het in kaart brengen van gebieden en het bezorgen van pakketjes. Voor deze taken kunnen ofwel micro helikopters worden gebruikt, die stil kunnen hangen maar niet erg efficiënt zijn, ofwel vliegtuigen met een vaste vleugel, die efficiënt zijn maar niet stil kunnen hangen. Hybride microdrones combineren het stilhangen van de helikopter met de efficiëntie van een vliegtuig met vaste vleugel. De reden dat dit type microdrone nog niet wijdverspreid is, komt doordat de besturing erg complex is.

Deze thesis behandelt de toepassing van Incrementele Niet-lineaire Dynamische Inversie (INDI) voor de besturing van de stand en snelheid van hybride microdrones. Deze besturingsmethode is voorafgaand aan deze thesis nog nooit praktisch toegepast, wat de reden is dat deze thesis begint met een toepassing van INDI op een quadrotor microdrone, die gemakkelijker te besturen is dan een hybride microdrone.

Allereerst wordt er een structuur voorgesteld voor de besturing van de hoekversnelling van een quadrotor. Ik laat zien dat de vertraging die het gevolg is van de filtering van het hoekversnellingssignaal, ook toegepast moet worden op de toestandsterugkoppeling van de actuator. Als hieraan voldaan wordt, komt het gebruikte filter niet voor in de overdrachtsfunctie van virtueel stuurcommando naar hoekversnelling, welke gelijk blijkt te zijn aan de dynamica van de actuator. Er wordt ook aangetoond dat verstoringen en niet gemodelleerde dynamica gecompenseerd worden met een overdrachtsfunctie die gelijk staat aan de dynamica van de actuator, vermenigvuldigd met het toegepaste filter en een stap vertraging. Tot slot wordt ook gedemonstreerd hoe het effect van de massatraagheid van de propeller, dat zeer significant kan zijn rond de gieras, aangepakt kan worden en hoe de besturingseffectiviteit adaptief gemaakt kan worden.

Ten tweede wordt in deze thesis een gewogen kleinste kwadraten besturingsallocatie algoritme aan de INDI besturing toegevoegd, teneinde de prioriteiten van de besturing te beheren. Dit betekent dat voor microdrones met gekoppelde sturingsmechanismen, bepaalde besturingsdoelen prioriteit gegeven kan worden wanneer saturatie van actuatoren zich voordoet. Dit is erg belangrijk voor microdrones met bestuurde assen die niet zo belangrijk zijn voor de stabiliteit van het toestel, zoals het gieren van een quadrotor. Het wordt gedemonstreerd dat, voor een quadrotor die een rotatie van 50 graden rond de gieras uitvoert, de stabiliteit sterk verbetert wanneer aan de gieras een lage prioriteit toegewezen wordt.

Ten derde introduceert deze thesis de besturing van lineaire versnellingen in alle drie de assen met INDI. Het algoritme heeft geen complex model nodig, en maakt in plaats daarvan gebruik van een meting van de versnelling van het toestel. Het wordt door middel van een windtunnel experiment aangetoond dat de robuustheid tegen verstoringen, die was aangetoond voor de besturing van de stand, ook van toepassing is op de besturing van lineaire versnellingen. Het wordt ook gedemonstreerd dat de methode buiten toegepast kan worden met een standaard GPS ontvanger. Tenslotte wordt een niet-lineaire methode voor het berekenen van het increment in stuurcommando afgeleid, wat slechts een kleine verbetering in het volgen van agressieve versnellingscommando's tot gevolg heeft.

Deze drie ontwikkelingen worden gecombineerd voor de besturing van hybride microdrones met INDI. Het resultaat is een enkele, continue INDI besturingsmethode voor de stand, en een enkele, continue INDI besturingsmethode voor de snelheid van het toestel. Dit wordt bereikt door het toevoegen van partiële afgeleides van de lift vector in de besturingseffectiviteit van de stamp- en rolhoek. Hoewel een transitie manoeuvre niet expliciet gedefinieerd is, volgt deze impliciet uit de incrementen die op de stand en de stuwkracht toegepast worden om gewenste versnellingsveranderingen te bewerkstelligen. Daarnaast wordt de gemodelleerde besturingseffectiviteit als functie van de vliegsnelheid aangepast, omdat de werkelijke besturingseffectiviteit sterk verandert binnen de vluchtbegrenzingen. Wanneer de vliegsnelheid te laag is om gemeten te kunnen worden, wordt de besturingseffectiviteit als functie van de stamphoek aangepast. Om (dwars) slippend vliegen te voorkomen is een sliphoekregelaar toegevoegd, waarbij een schatting van de sliphoek wordt gemaakt op basis van de versnellingsmeter.

Testvluchten tonen aan dat de INDI regelaars inderdaad in staat zijn om de stand en de lineaire versnelling van het toestel te besturen, binnen de vluchtbegrenzingen en binnen de fysische beperkingen van het vliegtuig. Het wordt ook aangetoond dat het volgen van versnellingscommando's het toestel op natuurlijke wijze over kan laten gaan naar een voorwaartse vlucht en terug, en overtrokken kan laten vliegen indien nodig. Door de abstractie die de INDI versnellingsregelaar met zich meebrengt, kan het toestel op eenvoudige wijze een snelheidsvectorveld volgen, bijvoorbeeld een veld dat het toestel een lijn laat volgen.

De ontwikkelde regelaar kan relatief gemakkelijk op andere hybride microdrones die op hun staart landen toegepast worden, omdat de modelafhankelijkheid laag is. Het algoritme zou zelfs, met kleine aanpassingen, op andere types hybride toestellen toegepast kunnen worden, zoals quadplanes of vliegtuigen met een roterende vleugel.

1

Introduction

In recent years, there has been a surge in the development of Micro Air Vehicles (MAVs). Most notably, the quadrotor (Figure 1.1a) has inspired many projects and applications, due to its mechanical simplicity, small form factor and hovering capability. Ever smaller electronics have allowed these vehicles to become smaller and more capable, giving them the potential to perform useful tasks in society. These tasks include aerial photography, inspection, search and rescue, and package delivery.

The application that inspired this thesis, is the placement of sensors in seismic data acquisition. Seismic measurements are very large operations, aimed at mapping different layers of the subsurface in an area of many square kilometers [1]. To this end, hundreds of thousands of accelerometers pick up the reflected vibrations from one or more sources. All these data can be processed by algorithms, in order to reconstruct the properties of different layers of the subsurface. Finally, the information obtained from such a seismic measurement can be used to find underground resources, such as oil, or to monitor existing resource reservoirs.

Manual placement of this many sensors in a very large area takes a lot of manpower. Especially in rough terrain, this can be a costly undertaking. Through the use of MAVs, the sensors could be delivered to the desired locations at a much lower price, compared to manual placement. However, fixed wing MAVs are not feasible for this task, because they can not carefully place a sensor at the correct location. On the other hand, rotorcraft MAVs do not have the required range and speed.

Hybrid MAVs combine hovering capability with fast and efficient wing-borne flight (Figure 1.1b). They generally consist of a wing, and an additional means of providing lift and control authority during hover flight. The wing gives the hybrid its range and endurance, as it is much more efficient at providing lift than a rotor. In order to maximize the endurance, a hybrid MAV has to spend most of its time in forward flight. Therefore, the vehicle should be designed to be as efficient as possible in forward flight, even if this comes at the cost of hovering efficiency. This also means that any additional weight that is not needed for forward flight should be avoided. Accordingly, to minimize the weight the hover flight should ideally be facilitated by the same propulsion system as the one that is used for forward flight.

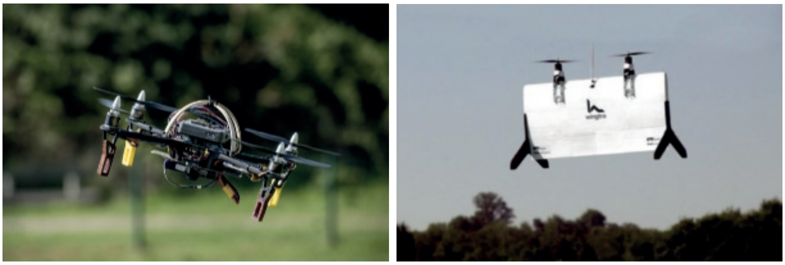

(a) (b)

Figure 1.1: A quadcopter (a) [2] and a tailsitter (b) [3].

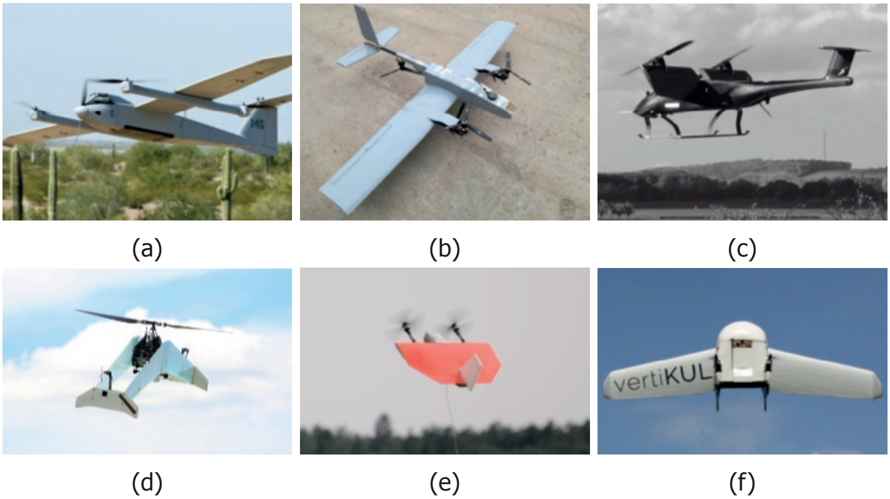
However, controlling the vehicle in the hover flight phase is generally more difficult using a propulsion system dedicated to forward flight. Fixed wing MAVs typically make use of flaps to generate control moments, which is effective if there is enough air blowing over them. While hovering, flaps may remain useful for control if they are located in the propeller slipstream. However, much stronger control moments can be generated with off-centered propellers, or by swash plate actuated rotors (control mechanism used by helicopters). Moreover, hybrid MAVs are more prone to being disturbed by wind gusts than other hovering vehicles. This is because of their wing surface, which increases their surface-to-mass ratio, leading to higher disturbing forces for the same vehicle mass.

Hence, the potential high gain of using hybrid MAVs is rivaled by the significant design and control challenges accompanying such vehicles. In what follows, we will discuss different existing types of hybrid MAVs and further flesh out the challenges involved.

1.1. Types of hybrid MAVs

There are many different possible designs of hybrid MAVs, such as tailsitters, quadplanes, tilt-rotors and tilt-wings. Figure 1.2 shows several examples of hybrid MAV designs, each demonstrating a different concept. First, Figure 1.2a displays a quadplane [4], which is arguably the most simple hybrid design. It consists of a traditional fixed wing MAV, with attached to it four rotors in a quadrotor configuration. A separate propeller provides the thrust in forward flight. Similar is the tilt-rotor, of which an example is shown in Figure 1.2b [5]. Here, the forward two propellers have a servo powered tilting mechanism, which allows them to provide lift during hover, as well as thrust in forward flight.

Figure 1.2c shows a tilt-wing, developed by Hartmann *et al.* [6] in combination with DHL. The propellers have a fixed pitch, and there is an additional rotor in the tail of the aircraft, which can provide pitching moments. Because of this rotor in the tail, the aircraft is able to perform trimmed flight at any airspeed.



(a) (b) (c)

(d) (e) (f)

Figure 1.2: Several different hybrid MAV designs: a quadplane (a) [4], a tilt-rotor (b) [5], a tilt-wing (c) [6] and tailsitters (d-f) [7–9]

A very different design is the Delftacopter [7], shown in Figure 1.2d. It is a tailsitter, with one big rotor equipped with a swash plate, which allows for pitch and roll moment generation. Small propellers are needed to counteract the torque from the main rotor. Further, it has a biplane wing configuration to minimize the span, which makes it less susceptible to wind gusts. The MAVion [8], shown in Figure 1.2e, is mechanically much simpler. It is also a tailsitter, but it has just two fixed pitch propellers, which provide airflow over the flaps, making them effective in hover as well as forward flight. This minimalistic design comes at the cost of controllability, as the moments that can be generated with the flaps in hover flight are generally small compared to moments that can be generated by off-centered propellers. Figure 1.2f shows the Vertikul [9], a tailsitter without flaps, which is only actuated by four propellers.

One way to categorize the different types of hybrids, is in designs that have their wings always aligned with the flow, and designs that fly at a very high angle of attack during a level transition. For example, quadplanes and tilt-rotors belong the first category, while tilt-wings and tailsitter belong to the second category. The benefit of designs from the first category is that the wing does not need to operate in complex stalled conditions. A downside is that either a mechanism to rotate the thrust vector, or a secondary thrust system is needed. This adds weight, and increases the chance of mechanical failure.

MAVs from the second category can only produce thrust parallel to the wing, such that the wing (but often the entire vehicle) has to rotate to accommodate hover flight. They can have a lower weight, as the same propulsion system is used in forward flight and hover. However, the downside of this single propulsion system, which is normally optimized for forward flight or hover, is that it now has to be some kind of compromise. Furthermore, this type of vehicle has to fly at very high angles of attack, which leads to a stalled turbulent flow over the wing. The stall makes it difficult to model the forces and moments, and the conditions under which the flow detaches and reattaches are hard to predict precisely.

1.2. Tailsitter hybrids

Of the different hybrid MAV types, one that is potentially very efficient is the tailsitter (Figure 1.1). This is for the simple reason that most tailsitters do not need any extra parts to facilitate hover flight. Instead, the vehicle itself tilts 90 degrees, such that the propellers that deliver thrust in forward flight also provide the lift during hovering flight. This saves weight, which increases the flight efficiency.

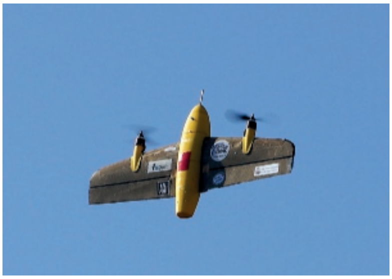


Figure 1.3: The Cyclone tailsitter.

One of the challenges that comes with such a design is the controllability. The transitions from hover to forward flight and back traverse very large angles of attack, which result in stalled flight conditions. This makes the forces and moments unpredictable and especially difficult to model, and as such it is a demanding control problem [10]. This is especially problematic as the vehicle may need to fly for prolonged periods of time in stalled conditions, for example when it needs to maintain position in the presence of wind.

Moreover, these hybrid vehicles have a large aerodynamic surface, orthogonal to horizontal wind gusts, which makes them very susceptible to wind gusts during hovering flight as compared to quadrotors or helicopters [9]. Since wind gusts are hard to predict, any model-based control approach will struggle with disturbance rejection performance. Google specified that poor gust disturbance rejection performance was one of the main reasons to terminate their tailsitter delivery drone concept [11].

In this thesis, I will investigate the control of the attitude and velocity of tailsitters. Nonetheless, the algorithms in this thesis could also be applicable to other types of hybrid MAVs. This is a little different for hybrid MAVs equipped with one or more large rotors with swash plates, which typically introduce additional dynamics, leading to a complex input-output relationship. This is beyond the scope of this thesis, and could be future research.

1.3. Control of hybrids

The control of hybrid MAVs can be divided into attitude control, velocity control and guidance. The sections below describe the research that has been carried out for each of these topics. It is not straightforward to compare control methods presented by different research groups, because they are typically applied to different vehicles, with different methods of actuation. Some of the most relevant work is not done on tailsitters, but on other types of hybrid vehicles. Nevertheless, I also discuss these, and attempt to compare the merits and downsides of these approaches.

1.3.1. Attitude control

Many approaches for attitude control of hybrids are discussed in literature. PID controllers are most prevalent, typically with some form of gain scheduling [5, 12–20]. The gain scheduling is necessary, because the control effectiveness can change a lot for the very different flight conditions of hover and forward flight, requiring different gains.

From these papers it becomes clear that, for the vehicles presented, it is possible to control the attitude with PID control. PID controllers have the benefit of being very easy to implement, and straightforward to tune. This has allowed the flight testing of many new hybrid concepts, without the requirement of an advanced model. The downside of PID control is that the performance is limited, especially for nonlinear systems subject to disturbances. This is mainly due to the fact that to arrive at zero tracking error, the integrator has to compensate aerodynamic moments, which it can do only slowly. The aerodynamic moments especially vary during the transition, when the vehicle undergoes large changes in angle of attack and airspeed, and the transition of the airflow between a stalled and an attached state.

If an aerodynamic dataset is available from wind tunnel experiments, aerodynamic moments can be compensated in a feedforward fashion [3, 21, 22]. Ideally, the aerodynamic moments, through this compensation, do not require any feedback from the PID controller, improving performance. It is also possible to use wind tunnel data to develop multiple linearized models, distributed over the flight envelope, as is done by Lustosa *et al.* [23]. For each of these models, they designed a Linear-Quadratic Regulator (LQR) that covers part of the flight envelope. Hartmann *et al.* [21] show that with an approach based on an aerodynamic model, good tracking of attitude angles can be achieved. Nevertheless, such a model is not easy to obtain, and often requires wind tunnel experiments. Moreover, modifications to the shape of the airframe would require redetermination of the aerodynamic parameters.

As an alternative to an accurate model of the vehicle along with aerodynamic data, some have implemented adaptive control schemes [24–27]. If the vehicle can adapt to the changing flight conditions while flying, an aerodynamic model is not needed. Successful flights have been demonstrated using this method, demonstrating the applicability of this concept. Still, it is hard to guarantee that such a system will not be influenced by disturbances, which may make it learn the wrong dynamics. This can lead to large parameter variation, which seems to be the case in the research of Knoebel *et al.* [25].

Finally, there has been some research into integrating multiple pressure sensors in the vehicle, such that gust disturbances can be measured and counteracted [28]. Experiments show that the pitch control can be improved using these sensors when subject to a gust disturbance. This benefit comes at the cost of added weight and increased complexity. Further, by relying on an extra sensor, an additional point of failure is introduced.

1.3.2. Velocity control

Once the attitude is controlled (inner loop), an outer loop controller needs to make sure that the vehicle goes where it is supposed to, by manipulating the attitude and the thrust. The dual capability of hover and forward flight invites the development of a modal controller, with modes for hover, forward flight and the transition between the two [15, 29–31]. The benefit of this approach is that the individual controllers can be quite simple and straightforward.

However, Stone *et al.* [32] point out that the combination of these simple controllers can be quite complex. Moreover, such a control structure works best if there are indeed only two relevant flight regimes: hover and forward flight. In practice, it turns out that there are plenty of situations where the vehicle should be able to fly at a slow speed, in between hover and forward flight. This is especially the case on windy days, when the drone can only hover at a certain position by having a nonzero speed with respect to the air. The ability to do this is essential during the landing on such a day.

A controller that is able to control the velocity across the entire flight envelope was developed by Hartmann *et al.* [6]. They let the outer control loop control the velocity, through manipulation of the acceleration. This results in a controller that is able to fly at any velocity in the flight envelope, and does not need to explicitly transition from one flight condition to the other. Nonetheless, this is achieved with an extensive trim model, obtained through wind tunnel tests. As was mentioned before, wind tunnel tests are expensive and time consuming. Furthermore, modeling of aerodynamic forces will help the control for nominal flight, but not in the case of unmodeled disturbances.

1.4. Incremental Nonlinear Dynamic Inversion

A control method that may be able to deal with these problems is Incremental Nonlinear Dynamic Inversion (INDI) [33–35]. Compared to well known Nonlinear Dynamic Inversion (NDI), it replaces the model of the system dynamics with sensor measurements. INDI has been proposed for the control of the angular acceleration of aircraft, through which the attitude can be controlled, though it was never applied in practice. In this case, a measurement of the angular acceleration is needed, which can be obtained from the gyroscope measurement using finite difference. As the angular acceleration is caused by the sum of control inputs and external moments, a desired angular acceleration can be achieved by incrementing the inputs, based on a control effectiveness matrix. This eliminates the need to model the aerodynamic moments on the vehicle, as they will be measured through the angular acceleration. Additionally, all disturbances are also measured through the angular acceleration, and as such they can be immediately counteracted by incrementing the inputs.

In the light of the drawbacks of current approaches to the control of hybrid MAVs, INDI could bring many benefits. Therefore, the problem statement of this thesis is formulated as follows:

> Can hybrid MAVs be successfully controlled with Incremental Nonlinear Dynamic Inversion?

Past research into INDI for aircraft has never been applied beyond simulation. One of the points where these simulations fall short are the actuator dynamics. Because of the actuator dynamics, measured angular accelerations cannot be counteracted instantly. Furthermore, these simulations do not contain the levels of noise that are realistic for an MAV. MAVs are typically strongly vibrating, which leads to sensor measurements which are contaminated by strong high frequency components. A high level of noise requires low-pass filtering, which in turn leads to a delay. This is especially problematic for the incremental approach, as the result of input increments needs to be observed in the output, before further increments are applied.

Therefore, the first challenge is to bring the INDI attitude control previously proposed by Sieberling *et al.* [35] in simulation to the real world, taking into account real-world properties such as actuator dynamics and sensor noise. This constitutes the first research question:

> **Research question 1**
>
> How can the attitude of an unmanned aircraft be controlled with incremental nonlinear dynamic inversion?

Another problem for the application of INDI to real vehicles is actuator saturation. For many MAV designs, each actuator has an effect on multiple axes, and the right control moments are achieved by an inversion step. This inversion step decouples the different controlled axes of roll, pitch, yaw and thrust. If the inversion step leads to actuator commands that cannot be achieved, the moments will not be correctly decoupled and it will be somewhat arbitrary which axes will not be controlled properly. Nonetheless, some axes are more important for stable flight than others, and the controller should take these priorities into account. For example, when the vehicle hovers, rotation around the thrust vector has the lowest priority, as it is unimportant for the positioning of the vehicle. Still, gusts are likely to disturb the vehicle along this axis, and counteracting these disturbances should not take precedence over maintaining stability, whenever these objectives are in conflict.

This leads to the second research question:

Research question 2

How can the algorithm deal with control priorities in the case of actuator saturation?

Finally, it is an open question if, and how, INDI can be used to control the velocity of an MAV. To achieve this, INDI would have to control the linear accelerations of the MAV, which would allow a simple linear controller to control the velocity. For this control loop, the same challenges exist as in the inner control loop: the modeling of the forces on the vehicle is complex, and the position of the vehicle is sensitive to gusts. Therefore, it naturally invites the application of INDI in this control loop as well, in which the measurement is the (linear) acceleration, and the inputs to the system are the vehicle's attitude and thrust. As this controller will use the attitude as an input, which is controlled in the inner loop, the interaction between the inner and outer INDI structure needs to be investigated.

Hence, the third research question is formulated as:

Research question 3

How can the velocity of an unmanned aircraft be controlled with incremental nonlinear dynamic inversion?

If the previous research questions are answered, all ingredients are there to fly a hybrid MAV with incremental control. This then culminates in the final research question:

Research question 4

How should incremental nonlinear dynamic inversion be adapted to apply it to hybrid unmanned air vehicles?

1.5. Outline

This thesis is organized as follows. In Chapter 2, it is investigated how INDI can be applied to the attitude control of a quadrotor MAV, with special consideration for the actuator dynamics and the effects of the necessary filtering. The reason to first apply the controller to a quadrotor MAV is that it is a simpler vehicle, which allows for some of the challenges of applying INDI to be tackled before the specifics of tailsitters are dealt with. Although a quadrotor MAV has only a subset of the characteristics of a hybrid tailsitter MAV, the principles developed in the chapter apply to tailsitters as well. The INDI controller resulting from this chapter is used in Chapter 3, which deals with the management of priorities for an MAV subject to actuator saturation. This is done through the incorporation of a quadratic cost function in the control allocation step, with different weights depending on the axis priorities. This resultant controller is used again in Chapter 4, in which an extra

INDI loop is added that controls the linear acceleration, and through this controls the position of the quadrotor. The algorithms developed in chapters 2 through 4 are combined in Chapter 5, where they are extended such that they can be applied to a tailsitter MAV. Test flights confirm the flexibility and performance of the flight controller. In Chapter 6 it is concluded that the developed INDI controller is able to control a tailsitter throughout its entire flight envelope, without accurately modeling the aerodynamics of the vehicle and without switching the control structure. Compared to other approaches, the developed controller is simple, requires little modeling and performs well against disturbances.

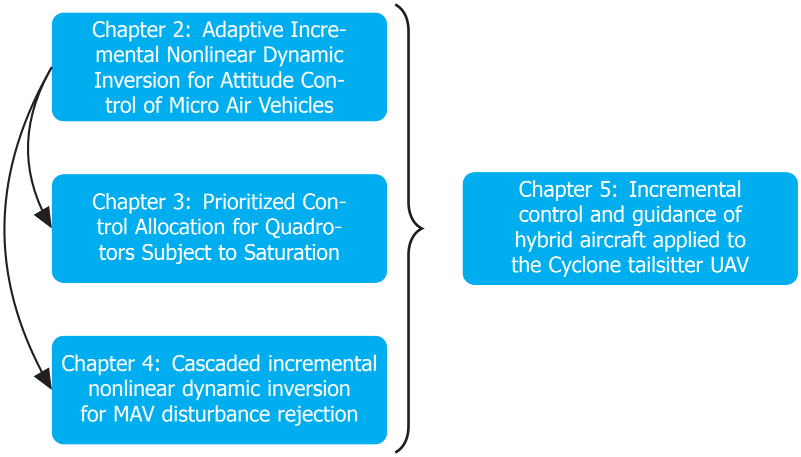

Chapter 2: Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles
Chapter 3: Prioritized Control Allocation for Quadrotors Subject to Saturation
Chapter 4: Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection
Chapter 5: Incremental control and guidance of hybrid aircraft applied to the Cyclone tailsitter UAV

Figure 1.4: Structure of the thesis.

2

Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles

In the introduction, it was explained that Incremental Nonlinear Dynamic Inversion (INDI) has great potential benefits for attitude control of Micro Air Vehicles (MAVs) in terms of reducing model dependence and disturbance rejection. However, past research has only been applied to simulations; whether it can be applied in practice remains an open question. Moreover, the effect of delays, caused by actuator dynamics and sensor noise filtering, is currently not known.

In this chapter, the application of INDI on a real quadrotor MAV is investigated and the effect of system delays is analyzed. It is derived what the input response of the controller is, and what the transfer function is for disturbance rejection. An online control effectiveness estimation algorithm is added to deal with changes in the control effectiveness during flight.

This chapter is based on the following article:
Smeur, E. J. J., Chu, Q. P., de Croon, G. C. H. E., March 2016. Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Aerial Vehicles. Journal of Guidance, Control, and Dynamics 39 (3), 450–461.

Nomenclature

b	=	Width of the vehicle, m
$\mathbf{I}_v$	=	Moment of inertia matrix of the vehicle, kg m^2
$\mathbf{I}_r$	=	Moment of inertia matrix of the rotor, kg m^2
$\mathbf{I}$	=	Identity matrix
i	=	Rotor index
k_1	=	Force constant of the rotors, kg m/rad
k_2	=	Moment constant of the rotors, kg m^2/rad
l	=	Length of the vehicle, m
$\mathbf{M}_a$	=	Aerodynamic moment vector acting on the vehicle, Nm
$\mathbf{M}_c$	=	Control moment vector acting on the vehicle, Nm
$\mathbf{M}_r$	=	Moment vector acting on the propeller, Nm
T_s	=	Sample time of the controller, s
$\mathbf{u}$	=	Actuator input vector, rad/s
$\mathbf{v}$	=	Vehicle velocity vector, m/s
$\boldsymbol{\mu}$	=	Adaptation rate diagonal matrix
$\boldsymbol{\Omega}$	=	Vehicle angular rate vector, rad/s
$\dot{\boldsymbol{\Omega}}$	=	Angular acceleration vector, rad/s^2
$\boldsymbol{\omega}$	=	Angular rate vector of the four rotors around the body z axis, rad/s
$\boldsymbol{\omega}_i$	=	Angular rate vector of rotor i around each of the body axes, rad/s

2.1. Introduction

Micro Aerial Vehicles (MAVs) have increased in popularity as low-cost lightweight processors and inertial measurement units (IMUs) have become available through the smartphone revolution. The inertial sensors allow stabilization of unstable platforms by feedback algorithms. Typically, the stabilization algorithm used for MAVs is simple Proportional Integral Derivative (PID) control [36, 37]. Problems with PID control occur when the vehicle is highly nonlinear or when the vehicle is subject to large disturbances like wind gusts.

Alternatively, we could opt for a model based attitude controller. A model based controller that can deal with nonlinear systems is nonlinear dynamic inversion (NDI), which involves modeling all of the MAV's forces and dynamics. Theoretically, this method can remove all nonlinearities from the system and create a linearizing control law. However, NDI is very sensitive to model inaccuracies [38]. Obtaining an accurate model is often expensive or impossible with the constraints of the sensors that are carried onboard a small MAV.

The incremental form of NDI, *Incremental* NDI or INDI, is less model dependent and more robust. It has been described in the literature since the late nineties [33, 39], sometimes referred to as *simplified* [40] or *enhanced* [41] NDI. Compared to NDI, instead of modeling the angular acceleration based on the state and inverting the actuator model to get the control input, the angular acceleration is *measured* and an *increment* of the control input is calculated based on a desired increment in angular acceleration. This way, any unmodeled dynamics, including wind gust disturbances, are measured and compensated. Since INDI makes use of a sensor measurement to replace a large part of the model, it is considered a *sensor based* approach.

INDI faces two major challenges. Firstly, the measurement of angular acceleration is often noisy and requires filtering. This filtering introduces a delay in the measurement, which should be compensated for. Secondly, the method relies on inverting and therefore modeling the controls. To achieve a more flexible controller, the control effectiveness should be determined adaptively.

Delay in the angular acceleration measurement has been a prime topic in INDI research. A proposed method to deal with these measurement delays is predictive filtering [35]. However, the prediction of angular acceleration requires additional modeling. Moreover, disturbances cannot be predicted. Initially, a setup with multiple accelerometers was proposed by Ostroff and Bacon [39] to measure the angular acceleration. This setup has some drawbacks, because it is complex and the accelerometers are sensitive to structural vibrations. Later, they discussed the derivation of angular acceleration from gyroscope measurements by using a second order filter [34]. To compensate for the delay introduced by the filter, Ostroff and Bacon use a lag filter on the applied input to the system. We show in this Chapter that perfect synchronization of input and measured output can be achieved by applying the filter used for the gyroscope differentiation on the incremented input as well.

Other research focused on compensating delays in the inputs by using a Lyapunov based controller design [42]. In this Chapter, we show that delayed inputs (actuator dynamics) are naturally handled by the INDI controller.

The control effectiveness is the sole model still required by INDI. The parameters can be obtained by careful modeling of the actuators and the moment of inertia, or by analyzing the input output data from flight logs. However, even if such a tedious process is followed, the control effectiveness can change during flight. For instance, this can occur due to changes in flight conditions [43] or actuator damage [44]. In order to cope with this, we propose a method to adaptively determine the control effectiveness matrices.

In this Chapter, we present three main contributions: (1) a mathematically sound way of dealing with the delays originating from filtering of the gyroscope measurements, (2) the introduction of an adaptive INDI scheme, which can estimate the control effectiveness online and (3) incorporation of propeller momentum in the controller design. These contributions are implemented and demonstrated on a Parrot Bebop quadrotor running the Paparazzi open source autopilot software. This is a commercially available quadrotor and the code is publicly available on Github[1].

The presented theory and results generalize to other vehicles in a straightforward manner. We have applied this control approach successfully to a variety of quadrotors. Some of these MAVs were able to measure the rotational rate of the rotors (actuator feedback), but some did not have this ability. The INDI controller is believed to scale well to different types of MAVs like helicopter, multirotor, fixedwing

[1] Reader can download the code at `https://github.com/EwoudSmeur/paparazzi` in the branch *bebop_indi_experiment*

or hybrid.

The outline of this Chapter is as follows. First, a model of the MAV will be discussed in Section 2.2. Second, Section 2.3 will deal with INDI and the analysis for this controller for a quadrotor. Section 2.4 is about the adaptive extension of INDI. Finally, in Section 2.5, the experimental setup is explained, followed by the results of the experiments in Section 2.6.

2.2. MAV Model

The Bebop quadrotor is shown in Figure 2.1 along with axis definitions. The actuators drive the four rotors, whose angular velocity in the body frame is given by $\boldsymbol{\omega}_i = [\omega_{i_x}, \omega_{i_y}, \omega_{i_z}]$, where i denotes the rotor number. The center of gravity is located in the origin of the axis system and the distance to each of the rotors along the X axis is given by l and along the Y axis by b.

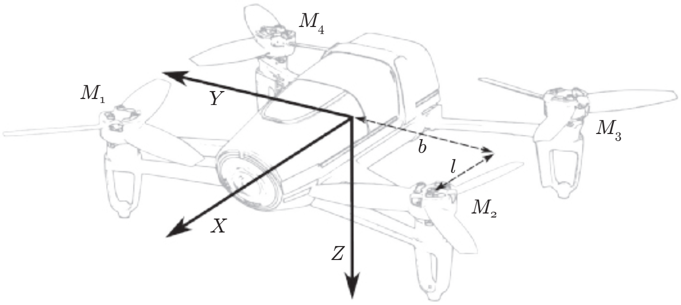


Figure 2.1: The Bebop Quadcopter used in the experiments with axis definitions.

If the angular velocity vector of the vehicle is denoted by $\boldsymbol{\Omega} = [p, q, r]^T$ and its derivative by $\dot{\boldsymbol{\Omega}}$, the rotational dynamics are given by Euler's equation of motion [45], more specifically the one that describes rotation. If we consider the body axis system as our coordinate system we get Eq. 2.1 for the angular velocity of the vehicle.

$$\mathbf{I}_v\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \mathbf{I}_v\boldsymbol{\Omega} = \mathbf{M} \tag{2.1}$$

Here $\mathbf{M}$ is the moment vector acting on the vehicle. If we consider the rotating propellers, still in the body coordinate system, we obtain:

$$\mathbf{I}_r\dot{\boldsymbol{\omega}}_i + \boldsymbol{\Omega} \times \mathbf{I}_r\boldsymbol{\omega}_i = \mathbf{M}_{r_i} \tag{2.2}$$

where $\boldsymbol{\omega}_i$ is the angular rate vector of the i^{th} propeller in the vehicle body axes and $\boldsymbol{\Omega}$ the angular rotation of the coordinate system, equal to the vehicle body rates. The rotors are assumed to be flat in the z axis, such that the inertia matrix $\mathbf{I}_r$ has elements that are zero: $I_{r_{xz}} = I_{r_{yz}} = 0$. Because the coordinate system is fixed to the vehicle, $I_{r_{xx}}$, $I_{r_{xy}}$ and $I_{r_{yy}}$ are not constant in time. However, as is shown later on, the terms containing these moments of inertia will disappear. Expanding

Eq. 2.2 into its three components gives:

$$\begin{aligned} I_{r_{xx}}\dot{\omega}_{i_x} - I_{r_{yy}}\Omega_z\omega_{i_y} - I_{r_{xy}}\Omega_z\omega_{i_x} + I_{r_{zz}}\Omega_y\omega_{i_z} &= M_{r_{i_x}} \\ I_{r_{yy}}\dot{\omega}_{i_y} + I_{r_{xx}}\Omega_z\omega_{i_x} + I_{r_{xy}}\Omega_z\omega_{i_y} - I_{r_{zz}}\Omega_x\omega_{i_z} &= M_{r_{i_y}} \\ I_{r_{zz}}\dot{\omega}_{i_z} - I_{r_{xx}}\Omega_y\omega_{i_x} - I_{r_{xy}}\Omega_y\omega_{i_y} + I_{r_{yy}}\Omega_x\omega_{i_y} + I_{r_{xy}}\Omega_x\omega_{i_x} &= M_{r_{i_z}} \end{aligned} \tag{2.3}$$

The propellers are light-weight and have a small moment of inertia compared to the vehicle. Relevant precession terms are therefore those that contain the relatively large ω_{i_z}. Since the rotors spin around the z axis, it is safe to assume that $\omega_{i_x} \ll \omega_{i_z}$ and $\omega_{i_y} \ll \omega_{i_z}$ and that $\dot{\omega}_{i_x}$ and $\dot{\omega}_{i_y}$ are negligible. Then, the moments exerted on the rotors due to their rotational dynamics are given by Eq. 2.4. Note the presence of the term $I_{r_{zz}}\dot{\omega}_{i_z}$, which is the moment necessary to change the angular velocity of a rotor. In Section 2.6, it will be shown that this term is important.

$$\mathbf{M}_{r_i} = \begin{bmatrix} M_{r_{i_x}} \\ M_{r_{i_y}} \\ M_{r_{i_z}} \end{bmatrix} = \begin{bmatrix} I_{r_{zz}}\Omega_y\omega_{i_z} \\ -I_{r_{zz}}\Omega_x\omega_{i_z} \\ I_{r_{zz}}\dot{\omega}_{i_z} \end{bmatrix} \tag{2.4}$$

This equation holds for each of the four rotors, so the moment acting on a rotor is given a subscript i to indicate the rotor number. The total moment due to the rotational effects of the rotors is shown in Eq. 2.5. Since motors 1 and 3 spin in the opposite direction of rotors 2 and 4, a factor $(-1)^i$ is introduced. As we are left with only the z component for the angular velocity of each rotor, we will omit this subscript and continue with the vector $\boldsymbol{\omega} = [-\omega_{1_z}, \omega_{2_z}, -\omega_{3_z}, \omega_{4_z}]^T = [\omega_1, \omega_2, \omega_3, \omega_4]^T$.

$$\begin{aligned} \mathbf{M}_r &= \textstyle\sum_{i=1}^4 \mathbf{M}_{r_i} = \sum_{i=1}^4 (-1)^i \begin{bmatrix} I_{r_{zz}}\Omega_y\omega_i \\ -I_{r_{zz}}\Omega_x\omega_i \\ I_{r_{zz}}\dot{\omega}_i \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -I_{r_{zz}} & I_{r_{zz}} & -I_{r_{zz}} & I_{r_{zz}} \end{bmatrix} \begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \\ \dot{\omega}_4 \end{bmatrix} + \begin{bmatrix} -I_{r_{zz}}\Omega_y & I_{r_{zz}}\Omega_y & -I_{r_{zz}}\Omega_y & I_{r_{zz}}\Omega_y \\ I_{r_{zz}}\Omega_x & -I_{r_{zz}}\Omega_x & I_{r_{zz}}\Omega_x & -I_{r_{zz}}\Omega_x \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \end{aligned} \tag{2.5}$$

Now consider the Euler Equation, Eq. 2.1, for the entire vehicle. The moments from the rotor dynamics are subtracted from the other moments yielding:

$$\mathbf{I}_v\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \mathbf{I}_v\boldsymbol{\Omega} = \mathbf{M}_c(\boldsymbol{\omega}) + \mathbf{M}_a(\boldsymbol{\Omega}, \mathbf{v}) - \mathbf{M}_r(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}, \boldsymbol{\Omega}) \tag{2.6}$$

Here, $\mathbf{I}_v$ is the moment of inertia matrix of the vehicle, $\mathbf{M}_r(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}, \boldsymbol{\Omega})$ is the gyroscopic effect of the rotors, $\mathbf{M}_c(\boldsymbol{\omega})$ is the control moment vector generated by the rotors and $\mathbf{M}_a(\boldsymbol{\Omega}, \mathbf{v})$ is the moment vector generated by aerodynamic effects, which depends on the angular rates and the MAV velocity vector $\mathbf{v}$. The control moment $\mathbf{M}_c(\boldsymbol{\omega})$ is elaborated in Eq. 2.7, based on a quadratic relationship of rotor rotational rate with the produced force and moment [36]. Here k_1 is the force constant of the rotors,

k_2 is the moment constant of the rotors and b and l are defined in Figure 2.1.

$$\mathbf{M}_c = \begin{bmatrix} bk_1(-\omega_1^2+\omega_2^2+\omega_3^2-\omega_4^2) \\ lk_1(\omega_1^2+\omega_2^2-\omega_3^2-\omega_4^2) \\ k_2(\omega_1^2-\omega_2^2+\omega_3^2-\omega_4^2) \end{bmatrix} = \begin{bmatrix} -bk_1 & bk_1 & bk_1 & -bk_1 \\ lk_1 & lk_1 & -lk_1 & -lk_1 \\ k_2 & -k_2 & k_2 & -k_2 \end{bmatrix} \boldsymbol{\omega}^2 \tag{2.7}$$

If we now take Eq. 2.6, insert Eqs. 2.4 and 2.7 and solve for the angular acceleration $\dot{\mathbf{\Omega}}$, we arrive at the following

$$\begin{aligned} \dot{\mathbf{\Omega}} &= \mathbf{I}_v^{-1}(\mathbf{M}_a(\mathbf{\Omega},\mathbf{v}) - \mathbf{\Omega}\times\mathbf{I}_v\mathbf{\Omega}) + \mathbf{I}_v^{-1}(\mathbf{M}_c - \mathbf{M}_r) \\ &= \mathbf{F}(\mathbf{\Omega},\mathbf{v}) + \tfrac{1}{2}\mathbf{G}_{1_q}\boldsymbol{\omega}^2 + T_s\mathbf{G}_2\dot{\boldsymbol{\omega}} - \mathbf{C}(\mathbf{\Omega})\mathbf{G}_3\boldsymbol{\omega} \end{aligned} \tag{2.8}$$

where $\mathbf{F}(\mathbf{\Omega},\mathbf{v}) = \mathbf{I}_v^{-1}(\mathbf{M}_a(\mathbf{\Omega},\mathbf{v})-\mathbf{\Omega}\times\mathbf{I}_v\mathbf{\Omega})$ are the forces independent of the actuators and $\mathbf{G}_{1_q}$, $\mathbf{G}_2$, $\mathbf{G}_3$ and $\mathbf{C}(\mathbf{\Omega})$ are given by Eqs. 2.9, 2.10, 2.11 and 2.12 respectively. Note that, though the model this far is in continuous time, eventually the controller has to be applied to a discrete system. To ease future calculations, the sample time T_s of the control system of the quadrotor is introduced in matrix G_2 by multiplying with T_s/T_s. Also note that the minus of the subtraction of $\mathbf{M}_r$ is taken inside the matrix of $\mathbf{G}_2$, such that we have an addition in Eq. 2.8.

$$\mathbf{G}_{1_q} = 2\mathbf{I}_v^{-1}\begin{bmatrix} -bk_1 & bk_1 & bk_1 & -bk_1 \\ lk_1 & lk_1 & -lk_1 & -lk_1 \\ k_2 & -k_2 & k_2 & -k_2 \end{bmatrix} \tag{2.9}$$

$$\mathbf{G}_2 = \mathbf{I}_v^{-1}T_s^{-1}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ I_{r_{zz}} & -I_{r_{zz}} & I_{r_{zz}} & -I_{r_{zz}} \end{bmatrix} \tag{2.10}$$

$$\mathbf{G}_3 = \mathbf{I}_v^{-1}\begin{bmatrix} I_{r_{zz}} & -I_{r_{zz}} & I_{r_{zz}} & -I_{r_{zz}} \\ -I_{r_{zz}} & I_{r_{zz}} & -I_{r_{zz}} & I_{r_{zz}} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.11}$$

$$\mathbf{C}(\mathbf{\Omega}) = \begin{bmatrix} \mathbf{\Omega}_y & 0 & 0 \\ 0 & \mathbf{\Omega}_x & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.12}$$

Note that traditionally in the literature, the system solved by INDI has the form of $\dot{x} = f(x) + g(x,u)$ where x is the state of the system and u the input to the system. However, as becomes clear from Eq. 2.8, the quadrotor is actually a system of the form $\dot{x} = f(x) + g(x,u,\dot{u})$. In Section 2.3, a solution to this type of problem will be shown.

2.3. Incremental Nonlinear Dynamic Inversion

Consider Eq. 2.8 from the previous section. This equation has some extra terms compared to previous work [35], because the gyroscopic and angular momentum effects of the rotors are included. We can apply a Taylor expansion to Eq. 2.8 and if we neglect higher order terms this results in Eq. 2.13:

$$\begin{aligned}\dot{\boldsymbol{\Omega}} = \quad & \mathbf{F}(\boldsymbol{\Omega}_0, \mathbf{v}_0) + \tfrac{1}{2}\mathbf{G}_{1_q}\boldsymbol{\omega}_0^2 + T_s\mathbf{G}_2\dot{\boldsymbol{\omega}}_{\mathbf{0}} - \mathbf{C}(\boldsymbol{\Omega}_0)\mathbf{G}_3\boldsymbol{\omega}_0 \\ & + \tfrac{\partial}{\partial\boldsymbol{\Omega}}(\mathbf{F}(\boldsymbol{\Omega}, \mathbf{v}_0) + \mathbf{C}(\boldsymbol{\Omega})\mathbf{G}_3\boldsymbol{\omega}_0)|_{\boldsymbol{\Omega}=\boldsymbol{\Omega}_0}(\boldsymbol{\Omega} - \boldsymbol{\Omega}_0) \\ & + \tfrac{\partial}{\partial\mathbf{v}}(\mathbf{F}(\boldsymbol{\Omega}_0, \mathbf{v}))|_{\mathbf{v}=\mathbf{v}_0}(\mathbf{v} - \mathbf{v}_0) \\ & + \tfrac{\partial}{\partial\boldsymbol{\omega}}(\tfrac{1}{2}\mathbf{G}_{1_q}\boldsymbol{\omega}^2 - \mathbf{C}(\boldsymbol{\Omega}_0)\mathbf{G}_3\boldsymbol{\omega})|_{\boldsymbol{\omega}=\boldsymbol{\omega}_0}(\boldsymbol{\omega} - \boldsymbol{\omega}_0) \\ & + \tfrac{\partial}{\partial\dot{\boldsymbol{\omega}}}(T_s\mathbf{G}_2\dot{\boldsymbol{\omega}})|_{\dot{\boldsymbol{\omega}}=\dot{\boldsymbol{\omega}}_0}(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_0)\end{aligned} \tag{2.13}$$

This equation predicts the angular acceleration after an infinitesimal timestep ahead in time based on a change in angular rates of the vehicle and a change in rotational rate of the rotors. Now observe that the first terms give the angular acceleration based on the current rates and inputs: $\mathbf{F}(\boldsymbol{\Omega}_0, \mathbf{v}_0) + \frac{1}{2}\mathbf{G}_{1_q}\boldsymbol{\omega}_0^2 + T_s\mathbf{G}_2\dot{\boldsymbol{\omega}}_{\mathbf{0}} - \mathbf{C}(\boldsymbol{\Omega}_0)\mathbf{G}_3\boldsymbol{\omega}_0 = \dot{\boldsymbol{\Omega}}_0$. This angular acceleration can be obtained by deriving it from the angular rates, which are measured with the gyroscope. In other words, these terms are replaced by a sensor measurement, which is why INDI is also referred to as *sensor based* control.

The second and third term, partial to $\boldsymbol{\Omega}$ and $\mathbf{v}$, are assumed to be much smaller than the fourth and fifth term, partial to $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$. This is commonly referred to as the principle of time scale separation [46]. This assumption only holds when the actuators are sufficiently fast and have more effect compared to the change in aerodynamic and precession moments due to changes in angular rates and body speeds. These assumptions and calculation of the partial derivatives gives Eq. 2.14:

$$\dot{\boldsymbol{\Omega}} = \dot{\boldsymbol{\Omega}}_0 + \mathbf{G}_{1_q}\mathrm{diag}(\boldsymbol{\omega}_0)(\boldsymbol{\omega} - \boldsymbol{\omega}_0) + T_s\mathbf{G}_2(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_0) - \mathbf{C}(\boldsymbol{\Omega}_0)\mathbf{G}_3(\boldsymbol{\omega} - \boldsymbol{\omega}_0) \tag{2.14}$$

Above it is stated that the angular acceleration is measured by deriving it from the angular rates. In most cases, the gyroscope measurements from a MAV are noisy due to vibrations of the vehicle due to the propellers and motors. Since differentiation of a noisy signal amplifies the noise, some filtering is required. The use of a second order filter is adopted from the literature [34], of which a transfer function in the Laplace domain is given by Eq. 2.15. Other low pass filters are also possible, for instance the Butterworth filter.

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.15}$$

The result is that instead of the current angular acceleration, a filtered and therefore delayed angular acceleration $\dot{\boldsymbol{\Omega}}_f$ is measured. Since all the terms with the zero subscript in the Taylor expansion should be at the same point in time, they are all replaced with the subscript f, yielding Eq. 2.16. This indicates that these signals are also filtered and are therefore synchronous with the angular acceleration.

$$\dot{\boldsymbol{\Omega}} = \dot{\boldsymbol{\Omega}}_f + \mathbf{G}_{1_q}\mathrm{diag}(\boldsymbol{\omega}_f)(\boldsymbol{\omega} - \boldsymbol{\omega}_f) + T_s\mathbf{G}_2(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_f) - \mathbf{C}(\boldsymbol{\Omega}_f)\mathbf{G}_3(\boldsymbol{\omega} - \boldsymbol{\omega}_f) \tag{2.16}$$

This equation is not yet ready to be inverted, because it contains the derivative of the angular rate of the propellers. Since on any embedded system the signals in

Eq. 2.16 are discrete signals, consider the discrete approximation of the derivative in the z domain: $\dot{\boldsymbol{\omega}} = (\boldsymbol{\omega} - \boldsymbol{\omega} z^{-1})T_s^{-1}$, where T_s is the sample time. This is shown in Eq. 2.17:

$$\dot{\boldsymbol{\Omega}} = \dot{\boldsymbol{\Omega}}_f + \mathbf{G}_{1_q}\mathrm{diag}(\boldsymbol{\omega}_f)(\boldsymbol{\omega}-\boldsymbol{\omega}_f) + \mathbf{G}_2(\boldsymbol{\omega}-\boldsymbol{\omega}z^{-1}-\boldsymbol{\omega}_f+\boldsymbol{\omega}_f z^{-1}) - \mathbf{C}(\boldsymbol{\Omega}_f)\mathbf{G}_3(\boldsymbol{\omega}-\boldsymbol{\omega}_f) \tag{2.17}$$

Note that the angular acceleration vector is a discrete variable in this equation, such that $\dot{\boldsymbol{\Omega}}[k]$ is the angular acceleration at time kT_s. It has to be approximated by a finite difference differentiation of the gyroscope in any real application, except if the system has an angular acceleration sensor. Collecting all terms with $(\boldsymbol{\omega} - \boldsymbol{\omega}_f)$ yields Eq. 2.18:

$$\dot{\boldsymbol{\Omega}} = \dot{\boldsymbol{\Omega}}_f + (\mathbf{G}_{1_q}\mathrm{diag}(\boldsymbol{\omega}_f) + \mathbf{G}_2 - \mathbf{C}(\boldsymbol{\Omega}_f)\mathbf{G}_3)(\boldsymbol{\omega}-\boldsymbol{\omega}_f) - \mathbf{G}_2 z^{-1}(\boldsymbol{\omega}-\boldsymbol{\omega}_f) \tag{2.18}$$

Inversion of this equation for $\boldsymbol{\omega}$ yields Eq. 2.19, where $^+$ denotes the Moore-Penrose pseudoinverse:

$$\boldsymbol{\omega}_c = \boldsymbol{\omega}_f + (\mathbf{G}_{1_q}\mathrm{diag}(\boldsymbol{\omega}_f) + \mathbf{G}_2 - \mathbf{C}(\boldsymbol{\Omega}_f)\mathbf{G}_3)^+(\boldsymbol{\nu} - \dot{\boldsymbol{\Omega}}_f + \mathbf{G}_2 z^{-1}(\boldsymbol{\omega}_c - \boldsymbol{\omega}_f)) \tag{2.19}$$

Note that the predicted angular acceleration $\dot{\boldsymbol{\Omega}}$ is now instead a virtual control, denoted by $\boldsymbol{\nu}$. The virtual control is the desired angular acceleration, and with Eq. 2.19, the required inputs $\boldsymbol{\omega}_c$ can be calculated. The subscript c is added to $\boldsymbol{\omega}$ to indicate that this is the command sent to the motors. This input is given with respect to a previous input $\boldsymbol{\omega}_f$. If we define the increment in the motor commands as $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega}_c - \boldsymbol{\omega}_f$, it is clearly an *incremental* control law.

2.3.1. Parameter Estimation

Equation 2.19 shows the general quadrotor INDI control law. The parameters of this equation are the three matrices $\mathbf{G}_{1_q}$, $\mathbf{G}_2$ and $\mathbf{G}_3$ which need to be identified for the specific quadrotor. This can be done through measurement of each of the components that make up these matrices, including the moments of inertia of the vehicle and the propellers as well as the thrust and drag coefficients of the rotors. Identifying the parameters in this way requires a significant amount of effort.

A more effective method is to use test flight data to determine the model coefficients. Of course, to do this the MAV needs to be flying. This can be achieved by initially tuning the parameters. Alternatively, a different controller can be used at first to gather the test flight data, such as PID control. Once a test flight has been logged, Eq. 2.18 is used for parameter estimation and is written as Eq. 2.20. From this equation, a least squares solution is found for the matrices $\mathbf{G}_{1_q}$, $\mathbf{G}_2$ and $\mathbf{G}_3$.

$$\Delta\dot{\boldsymbol{\Omega}}_f = \begin{bmatrix} \mathbf{G}_{1_q} & \mathbf{G}_2 & \mathbf{C}(\boldsymbol{\Omega}_f)\mathbf{G}_3 \end{bmatrix} \begin{bmatrix} \mathrm{diag}(\boldsymbol{\omega}_f)\Delta\boldsymbol{\omega}_f \\ (\Delta\boldsymbol{\omega}_f - z^{-1}\Delta\boldsymbol{\omega}_f) \\ -\Delta\boldsymbol{\omega}_f \end{bmatrix} \tag{2.20}$$

Here, Δ denotes the finite difference between two subsequent samples. From the data, we can also investigate the importance of some of the terms by comparing the least squares error with and without the terms. It turns out that on a typical dataset, leaving out the matrix $\mathbf{G}_3$ only results in an estimation squared error increase of ~0.2%. Furthermore, modeling the rotor as linear with the rotational speed of the rotor instead of quadratic gives an estimation squared error increase of ~0.9%. Therefore, we can simplify the INDI control law of Eq. 2.19 to Eq. 2.21:

$$\boldsymbol{\omega}_c = \boldsymbol{\omega}_f + (\mathbf{G}_1 + \mathbf{G}_2)^+(\boldsymbol{\nu} - \dot{\boldsymbol{\Omega}}_f + \mathbf{G}_2 z^{-1}(\boldsymbol{\omega}_c - \boldsymbol{\omega}_f)) \tag{2.21}$$

2.3.2. Implementation

With the simplifications described in subsection 2.3.1, the final INDI control scheme is shown in Figure 2.2. The input to the system is the virtual control $\boldsymbol{\nu}$ and the output is the angular acceleration of the system $\dot{\boldsymbol{\Omega}}$. The angular velocity measurement from the gyroscope is fed back through the second order filter and the differentiation, and is subtracted from the virtual control to give the angular acceleration error $\dot{\boldsymbol{\Omega}}_{\text{err}}$. A unit delay z^{-1} is added in the system block, to represent the fact that an input requires time to produce an output. For the filter of Eq. 2.15, satisfactory results were obtained with $\omega_n = 50$ rad/s and $\zeta = 0.55$.

Since the matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ are not square, we take the pseudo inverse to solve the problem of control allocation, denoted by $^+$. The contents of the block 'MAV' are shown in Figure 2.3, because it allows the closed loop analysis in Section 2.3.3. In this diagram, $\mathbf{d}$ is a disturbance term that bundles disturbances and unmodeled dynamics.

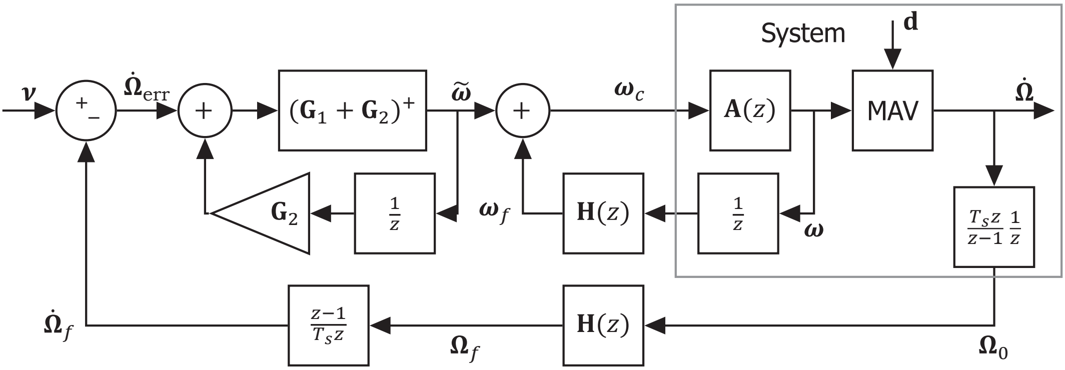


Figure 2.2: INDI control scheme. $\mathbf{A}(z)$ denotes the actuator dynamics and $\mathbf{H}(z)$ is the second order filter.

Note that Eq. 2.21 provides a desired angular velocity of the rotors. However, the actuators do not have an instantaneous response. Instead, it is assumed they have first order dynamics of the form:

$$A(z) = \frac{\alpha}{z - (1 - \alpha)} \tag{2.22}$$

where α is a constant. The reference sent to the motors is denoted by $\boldsymbol{\omega}_c$ and $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega}_c - \boldsymbol{\omega}_f$. In Figure 2.2, it is assumed that actuator feedback is available.

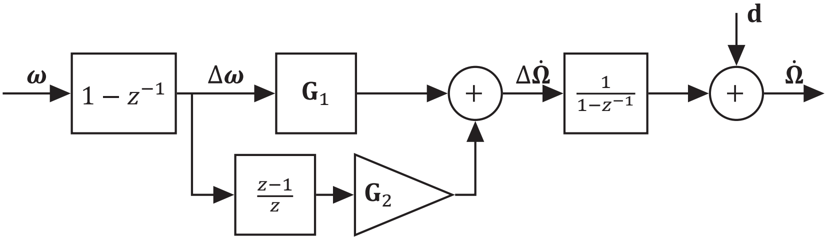


Figure 2.3: The contents of the block named 'MAV' in Figure 2.2.

However, if this is not the case, the actuator state $\boldsymbol{\omega}_0$ has to be estimated with a model of the actuator dynamics as is shown in Figure 2.4. Here $\mathbf{A}'(z)$ is a model of the actuator dynamics.

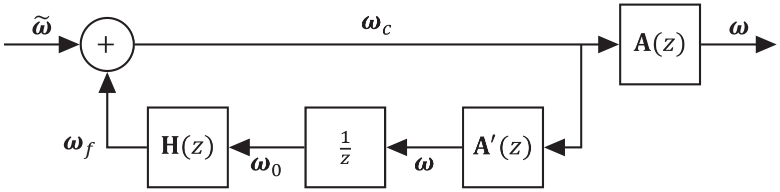


Figure 2.4: Block diagram for estimation of actuator state if actuator feedback is not available.

2.3.3. Closed Loop Analysis

Consider the control diagram shown in Figure 2.2. We can verify that this is a stable controller by doing a closed loop analysis. First, the transfer function of each of the two small loops is calculated, shown by Eq. 2.23 and 2.24. Here $\mathrm{TF}_{x\to y}$ denotes the transfer function from point x to y in the control diagram.

$$\begin{aligned}
\tilde{\boldsymbol{\omega}} &= (\mathbf{G}_1+\mathbf{G}_2)^{+}\dot{\boldsymbol{\Omega}}_{\mathrm{err}} + (\mathbf{G}_1+\mathbf{G}_2)^{+}\mathbf{G}_2 z^{-1}\tilde{\boldsymbol{\omega}}\\
(\mathbf{G}_1+\mathbf{G}_2)\tilde{\boldsymbol{\omega}} &= \dot{\boldsymbol{\Omega}}_{\mathrm{err}} + \mathbf{G}_2 z^{-1}\tilde{\boldsymbol{\omega}}\\
(\mathbf{G}_1+\mathbf{G}_2-\mathbf{G}_2 z^{-1})\tilde{\boldsymbol{\omega}} &= \dot{\boldsymbol{\Omega}}_{\mathrm{err}}\\
\mathrm{TF}_{\dot{\boldsymbol{\Omega}}_{\mathrm{err}}\to\tilde{\boldsymbol{\omega}}}(z) &= (\mathbf{G}_1+\mathbf{G}_2-\mathbf{G}_2 z^{-1})^{+}
\end{aligned} \tag{2.23}$$

We define $\mathbf{H}(z) = \mathbf{I}H(z)$ and assume that all actuators have the same dynamics, so $\mathbf{A}(z) = \mathbf{I}A(z)$. This means that each matrix in $\mathrm{TF}_{\tilde{\boldsymbol{\omega}}\to\boldsymbol{\omega}}(z)$ is a diagonal matrix and therefore $\mathrm{TF}_{\tilde{\boldsymbol{\omega}}\to\boldsymbol{\omega}}(z)$ is a diagonal matrix function.

$$\begin{aligned}
\mathrm{TF}_{\tilde{\boldsymbol{\omega}}\to\boldsymbol{\omega}}(z) &= (\mathbf{I}-\mathbf{A}(z)\mathbf{H}(z)z^{-1})^{-1}\mathbf{A}(z)\\
&= (\mathbf{I}-\mathbf{I}A(z)\mathbf{I}H(z)z^{-1})^{-1}\mathbf{I}A(z)\\
&= (\mathbf{I}(1-A(z)H(z)z^{-1}))^{-1}\mathbf{I}A(z)\\
&= \mathbf{I}(1-A(z)H(z)z^{-1})^{-1}A(z)
\end{aligned} \tag{2.24}$$

Then, the last part of the open loop is from $\boldsymbol{\omega}$ to $\dot{\boldsymbol{\Omega}}$, as shown by Figure 2.3. Using this figure, the transfer function is calculated in Eq. 2.25. Note that for this analysis, disturbances are not taken into account.

$$\mathrm{TF}_{\boldsymbol{\omega}\to\dot{\boldsymbol{\Omega}}}(z) = \mathbf{G}_1 + \frac{z-1}{z}\mathbf{G}_2 = \mathbf{G}_1 + \mathbf{G}_2 - \mathbf{G}_2 z^{-1} \tag{2.25}$$

Using these intermediate results, the open loop transfer function of the entire system is shown in Eq. 2.26:

$$\begin{aligned}\mathrm{TF}_{\dot{\boldsymbol{\Omega}}_{\mathrm{err}}\to\dot{\boldsymbol{\Omega}}}(z) &= \mathrm{TF}_{\boldsymbol{\omega}\to\dot{\boldsymbol{\Omega}}}(z)\mathrm{TF}_{\tilde{\boldsymbol{\omega}}\to\boldsymbol{\omega}}(z)\mathrm{TF}_{\dot{\boldsymbol{\Omega}}_{\mathrm{err}}\to\tilde{\boldsymbol{\omega}}}(z)\\ &= (\mathbf{G}_1 + \mathbf{G}_2 - \mathbf{G}_2 z^{-1})\mathbf{I}(1 - A(z)H(z)z^{-1})^{-1}A(z)(\mathbf{G}_1 + \mathbf{G}_2 - \mathbf{G}_2 z^{-1})^{+}\\ &= \mathbf{I}(1 - A(z)H(z)z^{-1})^{-1}A(z)\end{aligned} \tag{2.26}$$

Using Eq. 2.26 and Figure 2.2, we can calculate the closed loop transfer function of the entire system in Eq. 2.27:

$$\begin{aligned}\mathrm{TF}_{\boldsymbol{\nu}\to\dot{\boldsymbol{\Omega}}}(z) &= (\mathbf{I} + \mathrm{TF}_{\dot{\boldsymbol{\Omega}}_{\mathrm{err}}\to\dot{\boldsymbol{\Omega}}}(z)\mathbf{I}H(z)z^{-1})^{-1}\mathrm{TF}_{\dot{\boldsymbol{\Omega}}_{\mathrm{err}}\to\dot{\boldsymbol{\Omega}}}(z)\\ &= (\mathbf{I} + \mathbf{I}(1 - A(z)H(z)z^{-1})^{-1}A(z)\mathbf{I}H(z)z^{-1})^{-1}\mathbf{I}(1 - A(z)H(z)z^{-1})^{-1}A(z)\\ &= \mathbf{I}\frac{(1-A(z)H(z)z^{-1})^{-1}A(z)}{1+(1-A(z)H(z)z^{-1})^{-1}A(z)H(z)z^{-1}}\\ &= \mathbf{I}\frac{A(z)}{1-A(z)H(z)z^{-1}+A(z)H(z)z^{-1}}\\ &= \mathbf{I}A(z)\end{aligned} \tag{2.27}$$

From this equation, it appears that the closed loop transfer function from the virtual input to the angular acceleration is in fact the actuator dynamics $A(z)$. In most cases, the actuator dynamics can be represented by first or second order dynamics. Note that this shows the importance of applying the $H(z)$ filter on the input as well. By doing this, a lot of terms cancel and all that remains is the actuator dynamics.

Now, consider the transfer function from disturbances $\mathbf{d}$ (see Figure 2.2) to the angular acceleration. The derivation is given in Eq. 2.28 in which use is made of Eq. 2.26.

$$\begin{aligned}\mathrm{TF}_{\mathbf{d}\to\dot{\boldsymbol{\Omega}}}(z) &= (\mathbf{I} - \mathrm{TF}_{\dot{\boldsymbol{\Omega}}_{\mathrm{err}}\to\dot{\boldsymbol{\Omega}}}(z)(-1)\mathbf{H}(z)z^{-1})^{-1}\mathbf{I}\\ &= (\mathbf{I} + \mathbf{I}(1 - A(z)H(z)z^{-1})^{-1}A(z)\mathbf{I}H(z)z^{-1})^{-1}\mathbf{I}\\ &= \mathbf{I}\frac{1}{1+(1-A(z)H(z)z^{-1})^{-1}A(z)H(z)z^{-1}}\\ &= \mathbf{I}\frac{1-A(z)H(z)z^{-1}}{1-A(z)H(z)z^{-1}+A(z)H(z)z^{-1}}\\ &= \mathbf{I}(1 - A(z)H(z)z^{-1})\end{aligned} \tag{2.28}$$

With Eq. 2.28 we show that disturbances in the angular acceleration are rejected as long as the actuator dynamics and the designed filter are stable. The term $A(z)H(z)z^{-1}$ will go to 1 over time, with a response determined by the actuator dynamics, filter dynamics and a unit delay. This means that the faster the angular acceleration is measured, the faster the drone can respond and the faster the actuators can react, the faster the disturbance is neutralized.

2.3.4. Attitude Control

The angular acceleration of the MAV is accurately controlled by the system shown in Figure 2.2. To control the attitude of the MAV, a stabilizing angular acceleration reference needs to be passed to the INDI controller. This outer loop controller can be as simple as a Proportional Derivative (PD) controller (a gain on the rate error and a gain on the angle error), as shown in Figure 2.5. Here, $\boldsymbol{\eta}$ represents the attitude of the quadcopter. The benefit of the INDI inner loop controller is that the outer PD controller commands a reference, independent of the effectiveness of the actuators (including the inertia of the quadrotor).

This means that the design of this controller depends only on the speed of the actuator dynamics $\mathbf{A}(z)$. In case the actuator dynamics are known (through analysis of logged test flights for instance), a value of K_η and K_Ω can be determined that give a stable response.

This outer loop controller does not involve inversion of the attitude kinematics as has been done in other work [38]. However, the attitude angles for a quadrotor are generally small, in which case the inversion of the attitude kinematics can be replaced with simple angle feedback.

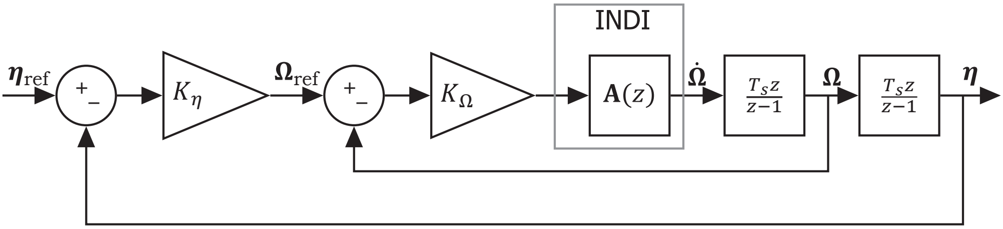


Figure 2.5: The design of the attitude controller based on the closed loop response of the INDI controller.

2.3.5. Altitude Control

The INDI controller derived in the beginning of this section controls the angular acceleration around the axes x, y and z, which corresponds to roll, pitch and yaw. However, there is a fourth degree of freedom that is controlled with the rotors, which is the acceleration along the z-axis.

Control of this fourth axis is handled by a separate controller. This controller scales the average input to the motors to a value commanded by the pilot, after the input has been incremented by the INDI controller.

2.4. Adaptive INDI

The INDI approach only relies on modeling of the actuators. The control effectiveness depends on the moment of inertia of the vehicle, the type of motors and propellers. A change in any of these will require re-estimation of the control effectiveness. Moreover, the control effectiveness can even change during flight, due to a change in flight velocity, battery voltage or actuator failure.

To counteract these problems and obtain a controller that requires no manual parameter estimation, the controller was extended with onboard adaptive parameter estimation using a Least Mean Squares (LMS) [47] adaptive filter. This filter is often used in adaptive signal filtering and adaptive neural networks.

The LMS implementation is shown in Eq. 2.29, where $\boldsymbol{\mu}_1$ is a diagonal matrix whose elements are the adaptation constant for each input and $\boldsymbol{\mu}_2$ is a diagonal matrix to adjust the adaptation constants per axis. This is necessary as not all axes have the same signal to noise ratio.

The LMS formula calculates the difference between the expected acceleration based on the inputs and the measured acceleration. Then it increments the control effectiveness based on the error. The control effectiveness includes both $\mathbf{G}_1$ as well as $\mathbf{G}_2$, as is shown in Eq. 2.30. Clearly, when there is no change in input, the control effectiveness is not changed. The reverse is also true: more excitation of the system will result in a faster adaptation. This is a benefit of the LMS algorithm over, for instance, recursive least squares with a finite horizon because recursive least squares will 'forget' everything outside the horizon.

$$\mathbf{G}(k) = \mathbf{G}(k-1) - \boldsymbol{\mu}_2\left(\mathbf{G}(k-1)\begin{bmatrix} \Delta\boldsymbol{\omega}_f \\ \Delta\dot{\boldsymbol{\omega}}_f \end{bmatrix} - \Delta\dot{\boldsymbol{\Omega}}_f\right)\begin{bmatrix} \Delta\boldsymbol{\omega}_f \\ \Delta\dot{\boldsymbol{\omega}}_f \end{bmatrix}^T \boldsymbol{\mu}_1 \tag{2.29}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 \end{bmatrix} \tag{2.30}$$

Note that the filtering can be different for the online parameter estimation than for the actual control. Equation 2.29 makes use of $\Delta\dot{\boldsymbol{\Omega}}_f$, which is the finite difference of $\dot{\boldsymbol{\Omega}}_f$ in the control Eq. 2.21. Since differentiating amplifies high frequencies, a filter that provides more attenuation of these high frequencies is necessary. We still use the second order filter described by Eq. 2.15, but with $\omega_n = 25$ rad/s and $\zeta = 0.55$.

When an approximate control effectiveness is given before takeoff, the adaptive system will estimate the actual values online, and thereby tune itself. The only knowledge provided to the controller is an initial guess of the control effectiveness. It is generally not possible to take off without any estimate of the control effectiveness, because the UAV might crash before the adaptive system has converged.

The choice of the adaptation constants $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ determines the stability and the rate of adaptation. By making these constants larger, a faster convergence is achieved. By making them too large, the adaptation will no longer be stable. The theoretical limit has been discussed in the literature [47] and it depends on the autocorrelation matrix of the input to the filter. In practice, the filter stability deteriorates before the theoretical limit, so in order to find a good adaptation constant some tuning is required.

2.5. Experimental Setup

To validate the performance of the INDI controller developed in Section 2.3 and the adaptive parameter estimation from Section 2.4, several experiments were conducted. These experiments were performed using the Bebop quadcopter from Parrot shown in Figure 2.1. The Bebop weighs 396.2 grams and can be equipped with bumpers, which are 12 grams per bumper. For these experiments, the bumpers where not equipped unless explicitly stated. The quadcopter was running the Paparazzi open source autopilot software, which contains all the code for wireless communication, reading sensor measurements etc. The accelerometer, gyroscope and control loops were running at 512 Hz.

Four experiments test the key properties of the controller:

- Performance
- Disturbance rejection
- Adaptation

During these experiments, the reference attitude and average thrust level were controlled by a pilot and sent to the drone over WiFi. All other computations were done on the drone itself, including the online adaptation.

2.5.1. Performance

In order to put the responsiveness of the system to the test and make sure that the angular acceleration reference is tracked by the INDI controller, a doublet input was applied on the attitude roll angle. The amplitude of the doublet is 30 degrees and the period is half a second (0.25 seconds positive and 0.25 seconds negative). This test is only done for the roll and not for the pitch, because there is no fundamental difference between these axes. The yaw axis is covered separately in experiment 2.5.4. Note that this experiment is performed without the adaptation.

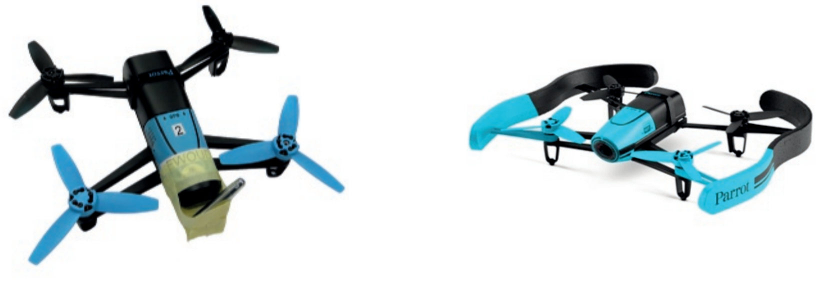
The performance is compared to a manually tuned PID controller. The INDI controller is not expected to be faster or slower than a traditional PID controller, because the result of Eq. 2.27 shows that the response of the INDI inner loop is simply the actuator dynamics. Considering that the outer loop is a PD controller, the rise time and overshoot should be similar.

Finally, this test will also be performed with an INDI controller that does not contain the filter delay compensation, so by using $\boldsymbol{\omega}_0$ in the controller increment instead of $\boldsymbol{\omega}_f$. It is expected that this will not fly well, because in Section 2.3.3 we showed that *with* this compensation all terms cancel and the closed loop transfer function reduces to $\mathbf{I}A(z)$.

By inspection of Figure 2.2, we can get a feel for what will happen if we omit this filter compensation. When there is an angular acceleration error, a control increment $\tilde{\boldsymbol{\omega}}$ will be the result, which is added to $\boldsymbol{\omega}_0$ to produce $\boldsymbol{\omega}_c$. $\boldsymbol{\omega}_c$ goes through the actuator dynamics to produce the new $\boldsymbol{\omega}$. The next time step, the result of this new $\boldsymbol{\omega}$ does not yet appear in $\dot{\boldsymbol{\Omega}}_f$, because it is filtered and therefore delayed. Therefore, $\tilde{\boldsymbol{\omega}}$ will be the same. However, $\boldsymbol{\omega}_0$ *did* update, so $\boldsymbol{\omega}_c$ will be incremented even more, while we are still waiting to see the result of the first increment in $\dot{\boldsymbol{\Omega}}_f$.

2.5.2. Disturbance Rejection

The disturbance rejection property is validated by adding a disturbance to the system. One possibility would be to apply aerodynamic disturbances by flying in the wake of a big fan. The disturbances occuring would be realistic, but not very repeatable. Moreover, the magnitude of the disturbance would be unknown.



(a) The container attached to the nose of the quadrotor with one weight inside.

(b) The Bebop quadrotor with bumpers.

Figure 2.6

Instead, it is possible to apply a disturbance in the form of a step function to the system. This is done by adding a weight of 42.5 grams to a container located in an off-centered position on the quadrotor while it is flying, as shown in Figure 2.6a. The container is located on the front of the drone and has a distance of about 11 cm to the center of gravity, so any weight added will shift the center of gravity forward. This will cause a misalignment of the thrust vector with respect to the center of gravity and therefore a pitch moment. This moment will be persistent and therefore have the form of a step disturbance. This is indicated with $\mathbf{d}$ in Figure 2.2. Although this moment is created with a center of gravity shift, the situation is the same as in the case of a persistent gust or an unmodeled aerodynamic moment.

A normal PID controller would respond to such a disturbance very slowly, because it takes time for the integrator to accumulate. But the introduction of the INDI inner loop leads to a cascaded control structure, which is much more resistant to disturbances than a single loop design [48]. Because of this, the reference pitch angle is expected to be tracked shortly after the disturbance.

2.5.3. Adaptation

The Bebop quadcopter has the possibility to fly with bumpers, as is shown in Figure 2.6b. Though these bumpers only weigh 12 grams a piece, they are located far from the center of gravity and therefore increase the moment of inertia. Furthermore, they can influence the airflow around the propellers. These system changes affect the $\mathbf{G}_1$ and $\mathbf{G}_2$ matrices. Therefore, the adaptive algorithm from Section 2.4 should deal with adding or removing the bumpers.

First, two flights are performed to show the effect of adding or removing the bumpers when the adaptive algorithm is not active. For the first flight, the bumpers are added, while the $\mathbf{G}_1$ and $\mathbf{G}_2$ matrices correspond to the quadrotor without bumpers. For the second flight, the bumpers are removed and the $\mathbf{G}$ matrices from the quadrotor with bumpers are used. In both flights, doublets are performed like in Section 2.5.1. The performance is expected to degrade compared to the previous results for both cases, as the $\mathbf{G}$ matrices do not correspond to what they should be.

Second, the ability of the quadrotor to adapt its $\mathbf{G}_1$ and $\mathbf{G}_2$ matrices is tested. In this experiment, the drone starts with bumpers equipped, but with system matrices that represent the configuration without bumpers. The pilot flies the drone in a confined area while performing some pitch, roll and yaw maneuvers to excite the system. While flying, the correct matrices should be estimated. Then, the Bebop is landed and the bumpers are removed. After take off, the matrices should converge to their original state.

Finally, doublets are performed with and without the bumpers equipped, while the adaptation algorithm is active. We expect the same performance as in Section 2.5.1.

These experiments are performed with the following values for $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$:

$$\boldsymbol{\mu}_1 = \mathbf{I}_8 \begin{bmatrix} 1 & 1 & 1 & 1 & 600 & 600 & 600 & 600 \end{bmatrix}^T \cdot 10^{-5} \tag{2.31}$$

$$\boldsymbol{\mu}_2 = \mathbf{I}_3 \begin{bmatrix} 1 & 1 & 0.3 \end{bmatrix}^T \tag{2.32}$$

where $\mathbf{I}_n$ is the nxn identity matrix. As was mentioned before, these values have been found through tuning. They can be lowered arbitrarily to slow down the adaptation, but can lead to instability of the adaptation when increased too much. On an offline dataset it was found that $\boldsymbol{\mu}_1$ could be increased with a factor of 30 before the adaptation became very volatile and close to unstable.

2.5.4. Yaw Control

The purpose of this experiment is to show the improvement in yaw performance due to the incorporation of the rotor spin-up torque in the controller design. This is done by applying a doublet input on the yaw setpoint. The amplitude of the doublet is 5 degrees and the period is one second (0.5 seconds positive and 0.5 seconds negative). As a comparison, the same experiment is performed with a traditional PID controller. This PID controller is manually tuned to give a fast rise time with minimal overshoot.

Additionally, the same test is performed with a zero $\mathbf{G}_2$ matrix. Here we expect an oscillation, because the persistent effect of a change in rotor angular velocity on the yaw axis is small. We take the pseudoinverse in Eq. 2.21, so the resulting gain will be very large. Because there is the angular momentum effect of the propellers, the initial angular acceleration will be larger than expected, and the controller will start to oscillate.

2.6. Results

This section deals with the results of the experiments described in Section 2.5. The angular acceleration shown in the plots in this section is not the onboard estimate of the angular acceleration, because it is delayed through filtering. Instead, it is computed after the experiment from the finite difference of the gyroscope data. The signal is filtered with a fourth order Butterworth filter with a cutoff frequency of 15 Hz. It is filtered twice, forward and reverse, resulting in a zero phase (noncausal) filter. For the actual control, the onboard filtered (and delayed) angular acceleration was used.

2.6.1. Performance

Figure 2.7 shows the angular acceleration around the x axis denoted by $\dot{p}$ and the reference angular acceleration denoted by $\dot{p}_{\text{ref}}$. Additionally, the reference is filtered with the actuator dynamics, resulting in $\dot{p}_{\text{ref}_A}$. This signal is the angular acceleration that is expected based on the calculations in Section 2.3.3, specifically Eq. 2.27. It might seem that the controller does not track the reference well because it lags behind the reference, but this was expected based on the model of the actuator dynamics. The angular acceleration is actually very close to the expected angular acceleration $\dot{p}_{\text{ref}_A}$. Finally, we also show the angular acceleration as calculated on board the quadrotor using the second order filter. The filtered angular acceleration on board the quadrotor is significantly delayed with respect to the actual angular acceleration, which is why we will run into problems if we don't take this delay into account in the INDI controller.

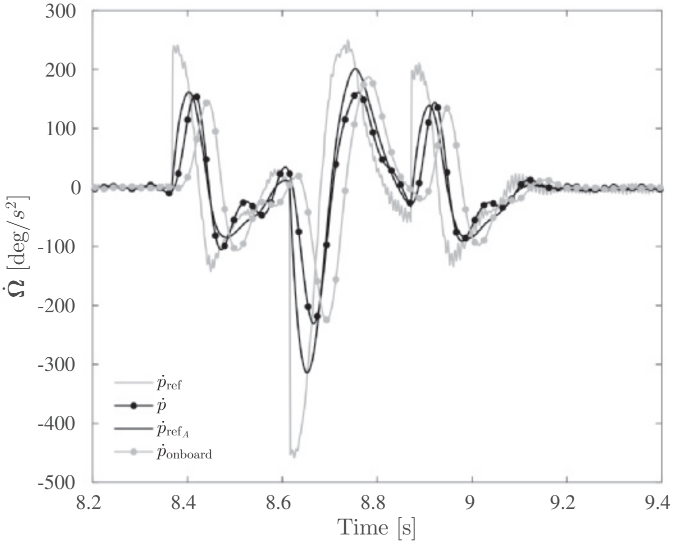


Figure 2.7: Angular acceleration in the roll axis during doublet input.

The outer loop controller, which generates the angular acceleration reference to track, was designed such that the resultant accelerations give a desired response of the roll angle, shown in Figure 2.8a. From this figure, it can be seen that the quadcopter reaches its reference roll angle within 0.2 seconds with a very small overshoot.

The roll angle response of the PID controller is shown in Figure 2.8b. As expected, the PID controller performs very similar to the INDI controller in terms of rise time and overshoot. The integral gain included in the PID controller, which needs to eliminate steady state offsets, degrades the dynamic performance of the

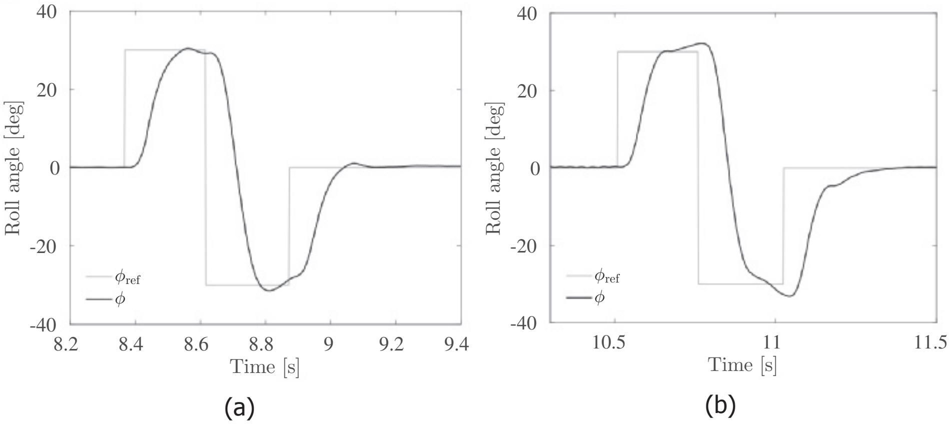


Figure 2.8: The roll angle during the doublet for INDI (a) and PID (b).

closed loop system. This shows that the INDI controller marginally improves the performance of a traditional PID controller in terms of responsiveness for the roll.

As discussed above, the onboard filtered measurement of the angular acceleration is significantly delayed. If we remove the filter delay compensation from the INDI controller, the quadrotor was severely oscillating, as can be seen in Figure 2.9. The doublet was not performed as this did not seem safe. The oscillation might be reduced by lowering K_η and K_Ω, but this will make the response slower as well. From this figure, we can conclude that the filter delay compensation is an important part of the INDI controller and is crucial in obtaining good performance with an INDI controller.

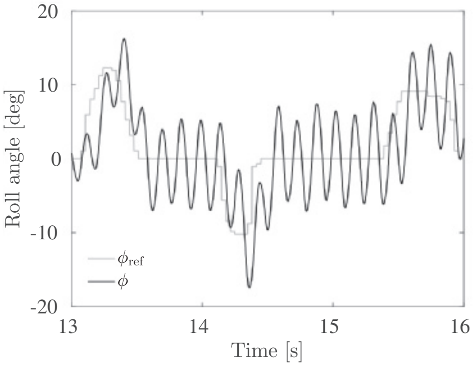


Figure 2.9: The roll angle for the INDI controller without filter compensation.

2.6.2. Disturbance Rejection

The weight, shown in Figure 2.6a, was placed in the container attached to the nose of the quadrotor by hand. The weight was placed in the container gently, but it probably arrived in the container with some small velocity. The disturbance in the angular acceleration is therefore a combination of a step and a delta pulse.

Figure 2.10 shows the angular acceleration that is the result of the disturbance. From the figure, it is clear that the disturbance happened just after 13 seconds. As the angular acceleration increases in the negative direction, the reference angular acceleration starts to go the opposite way, because now an angular rate and a pitch angle error start to arise. About 0.1 seconds after losing track of the reference, the angular acceleration again coincides with the expected angular acceleration, having overcome the disturbance in the angular acceleration.

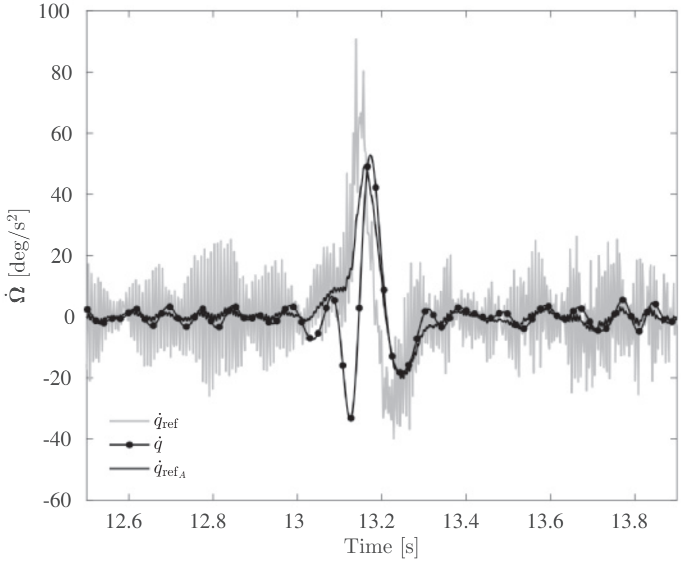


Figure 2.10: The angular acceleration during the disturbance.

This results in a pitch angle with no steady state error as can be seen from Figure 2.11. After 0.3 seconds, the pitch angle is back at zero. To show that the weight in the container really is a step disturbance, which can be compared to a constant aerodynamic moment, consider Figure 2.12. It shows the difference of the rotational rate of the front and rear motors divided by four: $(\omega_1 + \omega_2 - \omega_3 - \omega_4)/4$. This indicates the average magnitude in Rounds Per Minute (RPM) that each motor contributes to the pitch control (see Eq. 2.7). Clearly, there is a difference before and after the disturbance which can be quantified as an average change of 578 RPM over the interval [12.6 13.0] versus [13.4 13.8]. This demonstrates that the disturbance was really a step and that the INDI controller can rapidly cope with such a disturbance.

Figure 2.13 shows the same experiment performed with a PID controller. Of course, the weight was not dropped in exactly the same manner and with the same velocity, so the initial disturbance was probably different. However, the persisting disturbance is the same, because the weight has exactly the same mass. It takes

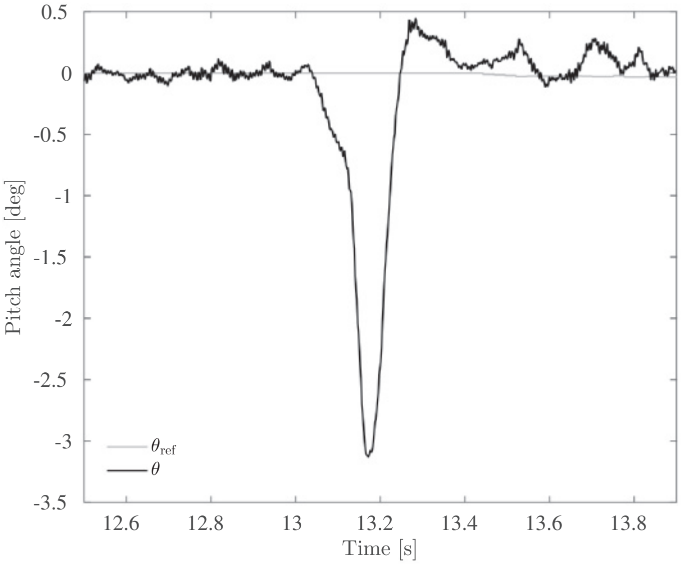


Figure 2.11: The pitch angle during the disturbance.

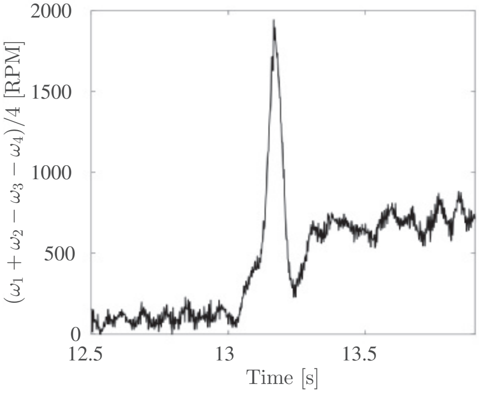


Figure 2.12: The difference between the rotational rate of the front motors and the rear motors.

about 1.5 seconds before the pitch angle is back at zero again, which is approximately 5 times longer than for the INDI controller. One might say that the integral gain of the PID controller should be larger, but this will deteriorate the performance in the previous experiment.

2.6.3. Adaptation

Figures 2.14 and 2.15 show the response to a roll doublet without adaptation if there is a mismatch in the control effectiveness. Even though the bumpers are lightweight, their effect is significant because they are located far from the center

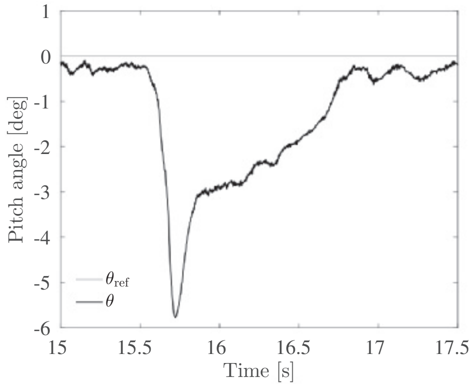


Figure 2.13: The pitch angle during the disturbance for the PID controller.

of gravity. In Figure 2.14, we see what happens if the actuators are less effective than in the model, because the inertia is higher. Additional increments of the input are needed to reach a desired angular acceleration. The oscillation occurs because this takes more time. The oscillation can be reduced by reducing the K_η and K_ω gains, at the cost of having a slower response.

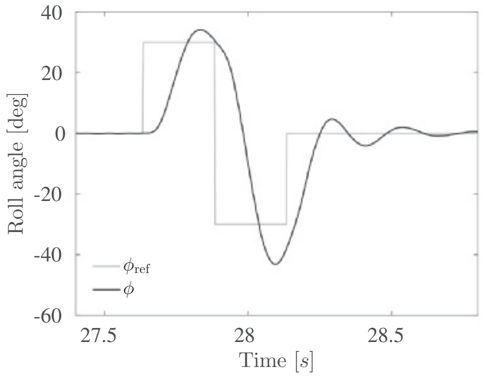


Figure 2.14: Flight without adaptation, with bumpers equipped, while the control effectiveness has been determined without bumpers

In Figure 2.15, we see the opposite: the control effectiveness is higher than what was modeled. This results in a fast oscillation, which *cannot* be removed by reducing the attitude gains. This is because the cause of the oscillation is different: now too much input is applied to reach a certain angular acceleration. This will happen regardless of what angular acceleration is requested by the attitude controller.

We can conclude that the performance degrades when the modeled control effectiveness does not closely correspond to the actual control effectiveness. When the adaptation algorithm is enabled, Figures 2.16a through 2.16c show how each

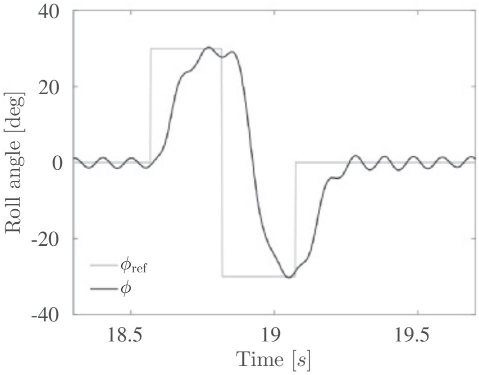


Figure 2.15: Flight without adaptation, without bumpers equipped, while the control effectiveness has been determined with bumpers

row of the $\mathbf{G}_1$ matrix evolves over time as a result of the second experiment described in section 2.5.3. The same is shown in Figure 2.16d for the third row of the $\mathbf{G}_2$ matrix. Each line represents one of the elements of that row, indicating the effectiveness of that motor on the specified axis.

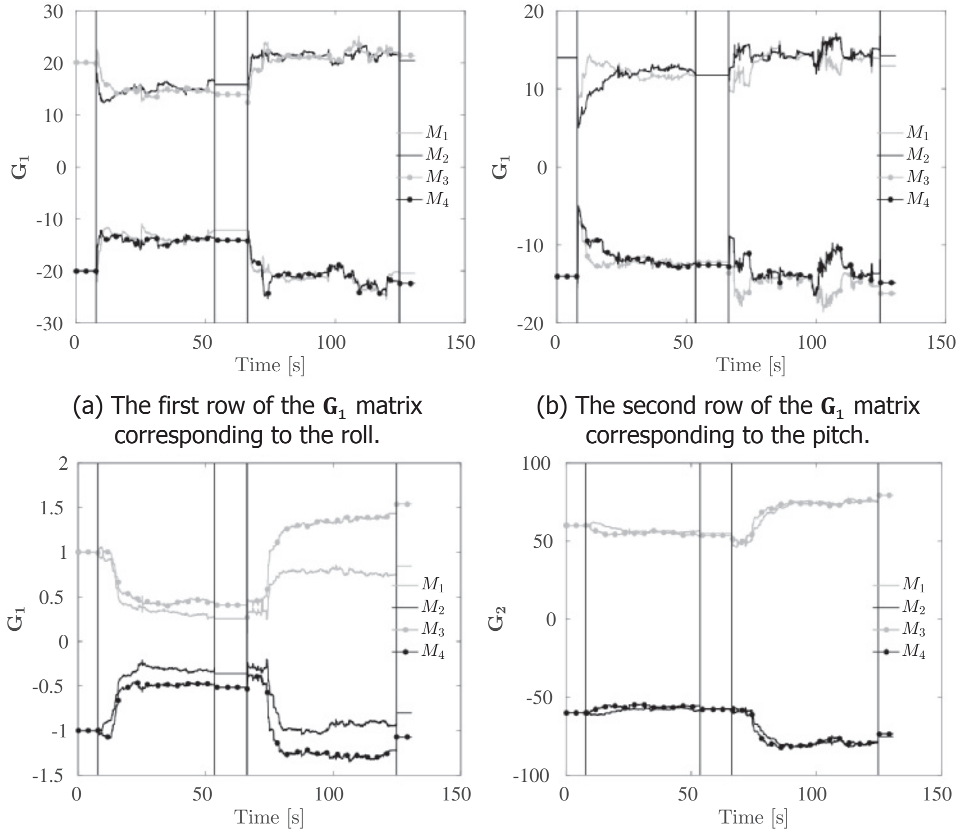
Note that the drone is flying in the interval of [8 54] seconds and again in [66 125] seconds; in between these times, the drone is landed and the bumpers are removed. This is indicated by vertical lines in the figures. A large change in effectiveness due to the addition and removal of the bumpers can be seen in the third row of the $\mathbf{G}_1$ matrix, shown in Figure 2.16c, which corresponds to the yaw.

Also in Figure 2.16a a change in effectiveness can be seen between the flights with and without bumpers. Once converged, the effectiveness values are stable with little noise. Upon take-off and landing the effectiveness seems to diverge for a short period of time. This is not a failure of the adaptation algorithm, but merely the result of the interaction with the floor.

The controller is engaged once the pilot gives a thrust command that exceeds idle thrust. At that point, the quadrotor does not produce enough lift to take off, so it is still standing on the floor. When the INDI controller tries to attain certain angular accelerations, the quadrotor does not rotate and the adaptation algorithm will adapt to this. When landing, these interactions with the floor can also occur.

Notice the large difference in effectiveness between the actuators in the second part of the flight in Figure 2.16c. This illustrates the added value of adaptive INDI, as often the actuators are assumed to perform equal to each other, while in this case they do not. These differences between the actuators are also observed with the estimation method described in subsection 2.3.1 for multiple flights. The differences may be caused by small imperfections that are not clearly visible on some of the rotors.

Finally, we can observe how the online parameter estimation affects the response to a roll doublet in Figures 2.17 and 2.18. Regardless of whether the bumpers are equipped or not or with what control effectiveness model the quadro-



(a) The first row of the $\mathbf{G}_1$ matrix corresponding to the roll.

(b) The second row of the $\mathbf{G}_1$ matrix corresponding to the pitch.

(c) The third row of the $\mathbf{G}_1$ matrix corresponding to the yaw.

(d) The third row of the $\mathbf{G}_2$ matrix corresponding to the yaw.

Figure 2.16: The adaptation experiment.

tor starts flying, the same performance is achieved as in Section 2.5.1. This shows the robustness of the adaptive algorithm against control effectiveness changes.

2.6.4. Yaw control

Finally, consider Figure 2.19. It shows for each timestep the change in angular acceleration in the yaw axis, $\Delta\dot{r}$, during the large control inputs discussed above. A careful reader up until this point may wonder: 'Is the rotor spin-up torque really significant? Can we not omit the $\mathbf{G}_2$ matrix?'. The figure shows the predicted change in angular acceleration based on the change in motor speeds according to Eq. 2.21, which is a close match. In green, the figure shows the predicted change in angular acceleration if we neglect $\mathbf{G}_2$, denoted by $\Delta\dot{r}_{\text{simple}}$. Clearly, the motor spin-up torque is very significant.

Moreover, if we try to fly with a zero $\mathbf{G}_2$ matrix, the resulting oscillation is so

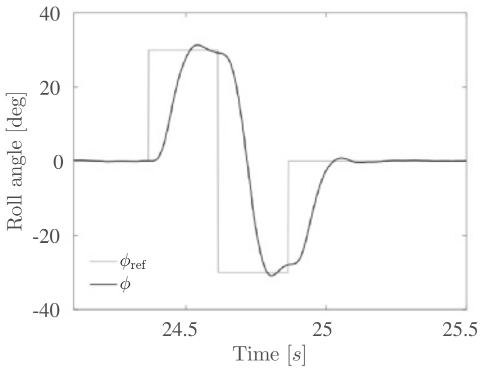


Figure 2.17: Flight with adaptation, with bumpers equipped, while the control effectiveness has been determined without bumpers

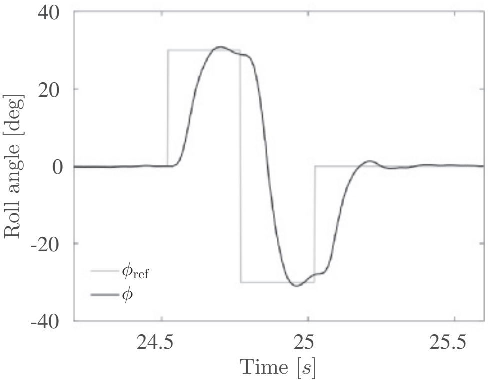


Figure 2.18: Flight with adaptation, without bumpers equipped, while the control effectiveness has been determined with bumpers

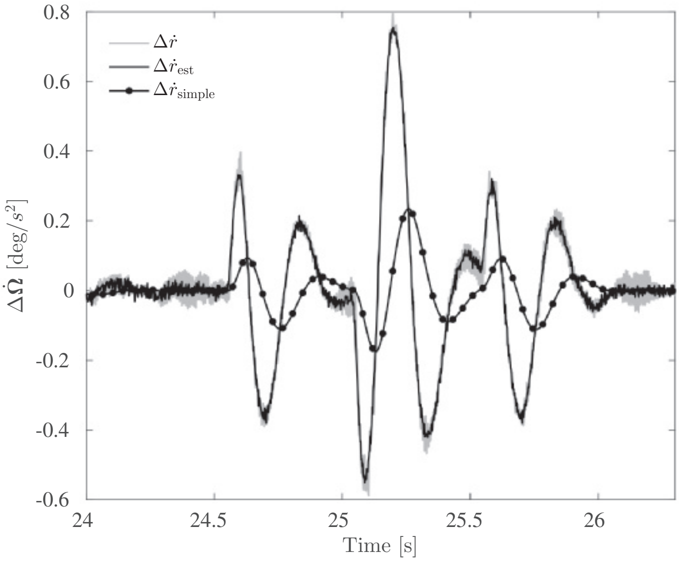


Figure 2.19: The change in angular acceleration in the yaw axis along with the predicted change.

strong that a takeoff is not possible. In order to fly without this matrix, we cannot use the estimated values for the control effectiveness in the yaw axis. Instead, we can take a higher effectiveness for the model parameters than in reality in order to avoid overshooting the reference angular acceleration due to the rotor spin-up torque that is now not taken into account. Figure 2.20 shows that it is possible to fly with a zero $\mathbf{G}_2$ matrix, at the cost of a severe performance penalty.

If we do take the rotor angular momentum into account, Figure 2.21a shows the resultant doublet response of the yaw angle. Compare this with Figure 2.21b,

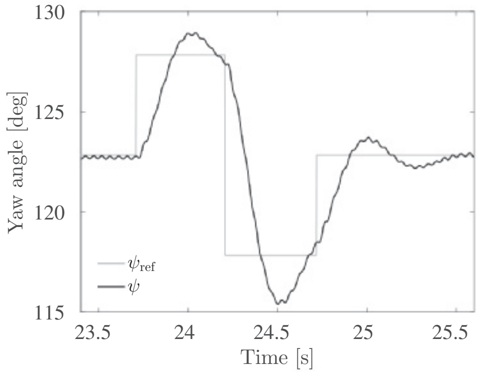
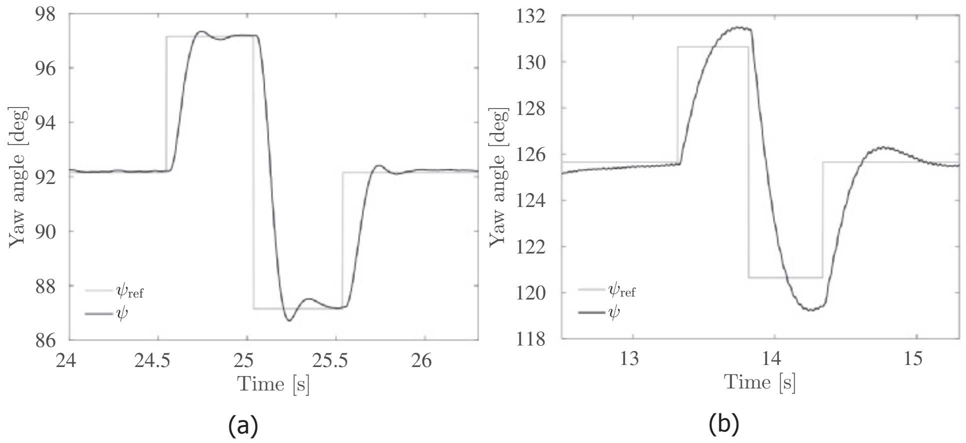


Figure 2.20: The yaw angle during the doublet for the INDI controller without $\mathbf{G}_2$ matrix.

which shows the doublet response for the PID controller. The INDI controller clearly has a faster rise time and less overshoot.



(a) (b)

Figure 2.21: The yaw angle during the doublet for INDI (a) and PID (b).

2.7. Conclusion

Adaptive Incremental Nonlinear Dynamic Inversion is a very promising technique for control of Micro Aerial Vehicles (MAV). Due to incorporation of the spin-up torque, fast yaw control is possible, which is typically very slow on a quadrotor. The disturbance rejection capabilities are vital when flying in windy conditions or with MAVs that have complex aerodynamics. Because unmodeled aerodynamic moments are measured with the angular acceleration, no complex aerodynamic modeling is needed. Even the control effectiveness matrices are shown to be adapted online, resulting in a controller that can handle changes in the MAV configuration and needs little effort to set up on a new platform. Only when a high performance outer loop is required is some knowledge of the actuator dynamics needed. These properties result in a very flexible and powerful controller.

3

Prioritized Control Allocation for Quadrotors Subject to Saturation

The previous chapter dealt with the application and analysis of incremental nonlinear dynamic inversion in practice. It was demonstrated through experiments that the vehicle behaved as expected from the theory in nominal cases. What was not yet taken into account was the possibility of actuator saturation, which can happen when performing demanding maneuvers. When actuators saturate, some or all of the calculated control inputs cannot be executed, which means that part of the control objective is not satisfied.

Since some control objectives are more important for the stability of the vehicle than others, this chapter provides a method that is able to allocate control effort depending on the priority of the different objectives. The method is applied to a quadrotor, and an experiment shows that the pitch and roll angles can be prioritized over yaw, improving the stability of the vehicle in demanding yaw tasks.

This chapter is based on the following conference paper:
Smeur, E. J. J., De Wagter, C., Höppener, D. C., September 2017. Prioritized Control Allocation for Quadrotors Subject to Saturation. International Micro Air Vehicle Conference and Flight Competition (IMAV), pp. 37-43.

3.1. Introduction

Control allocation is often described as the problem of distributing control effort over more actuators than the number of controlled variables [49–51]. This is something that occurs in traditional aircraft as well as drones, such as hexarotors and octorotors. What is also part of control allocation, is the problem of how to deal with actuator saturation, which can be very important in some cases.

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

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

In previous research, we have developed an Incremental Nonlinear Dynamic Inversion (INDI) controller for Micro Air Vehicles (MAV) [52, 53]. We have shown that this control method is very good at disturbance rejection and needs little model information. Moreover, we presented a method to include the effects of propeller inertia, yielding faster and more accurate yaw control. This aggressive yaw control can easily lead to saturation of multiple actuators, especially when commanding large yaw changes. These saturations often lead to errors in roll and pitch angles and in the thrust, causing the vehicle to lose control of its position and potentially crash.

But also external moments, such as wind disturbances, or actuator faults can lead to saturation. This is why a control allocation method needs to be added to the INDI control structure. Multiple control allocation algorithms have been proposed, some of which do not adequately address prioritization: ganging, redistributed pseudo-inverse, direct control allocation; and some of which do: linear programming and quadratic programming [54]. In this chapter, we will consider a quadratic cost function, and the corresponding quadratic optimization problem. A solution to this problem can be found in a straightforward way using the active set method, as has been shown by Härkegård [55].

As opposed to our approach of prioritization, some research has focused on the preservation of control *direction* [51, 56]. This means that in case of saturation, a solution for the actuator inputs is sought that corresponds to a linear scaling of the original control objective. This approach may be useful for systems where all axes are equally important. However, for a quadrotor, if a large yawing moment is needed, the actuators can easily saturate due to the low control effectiveness in this axis. Scaling the desired control moments will make the roll and pitch control suffer, which may lead to instability.

Recently, Faessler et al. implemented a heuristics based algorithm for priority management [57]. They showed that prioritizing roll and pitch over yaw can lead to stability improvements. However, the suggested algorithm resembles the Redistributed Pseudo Inverse method (RPI), which is known in some cases to not find the control solution even if the control objective is achievable [58]. Furthermore, the scheme is particularly constructed for quadrotors, and does not generalize.

The WLS approach is much more general, as it does not depend on a certain configuration of actuators. The method has been suggested for quadrotors by Monteiro *et al.* [59], but was only implemented in simulation. Furthermore, the weighting matrix, that determines the priorities in the cost function, is not discussed.

In this chapter, we integrate the Weighted Least Squares (WLS) control allocation algorithm into the INDI attitude controller. Further, we show through an experiment that prioritization of roll and pitch over yaw leads to stability improvements. The structure of this chapter is as follows: first, the INDI control law is introduced in Section 3.2. Second, Section 3.3 elaborates on the WLS method and how it integrates with INDI attitude control. Third, the experimental results are presented in Section 3.4, and we end with conclusions and future work in Section 3.5.

3.2. Incremental Nonlinear Dynamic Inversion and Actuator Saturation

In previous work [52], we derived INDI control for MAVs. A detailed derivation is beyond the scope of this chapter, but the main feature of the controller is its incremental way of controlling angular accelerations. The basic idea is that the current angular acceleration is caused by the combination of inputs and external moments. In order to change the angular acceleration, all that is needed is to take the previous inputs and increment them, based on the error in angular acceleration and the control effectiveness.

A distinction is made between two types of forces and moments: those that are produced by inputs, and those that are produced by changes in inputs. The forces and moments produced due to the propeller aerodynamics fall in the first category, and the torque it takes to spin up a propeller falls in the second category. Both need to be accounted for in different ways, which is why the control effectiveness matrix is split up in two parts: $G = G_1 + G_2$, where G_2 accounts for the propeller spin up torque. Though the algorithms presented here have broad applicability, we will, in order to promote clarity, consider the quadrotor shown in Figure 3.1, with the illustrated axis definitions. We define the angular rotation vector Ω, its derivative $\dot{\Omega}$ and the angular rate of the propellers ω. Then, if we assume a linear control effectiveness and that gyroscopic effects of the vehicle can be neglected [52], the system equation in incremental form is

$$\dot{\Omega} - \dot{\Omega}_0 + G_2 L(\omega - \omega_0) = (G_1 + G_2)(\omega - \omega_0), \tag{3.1}$$

subject to

$$\omega_{\text{min}} \leq \omega \leq \omega_{\text{max}}, \tag{3.2}$$

where L is the lag operator, e.g. $\omega(k-1) = L\omega(k)$. Note that the angular acceleration needs to be obtained by deriving it from gyroscope measurements through

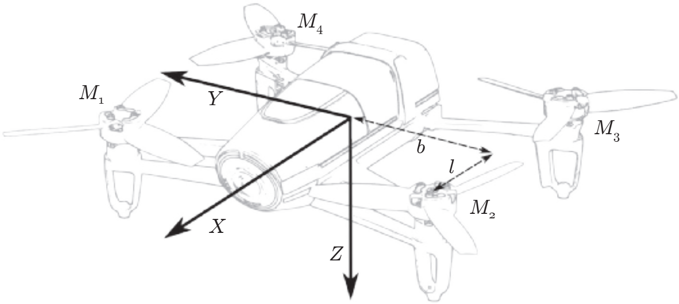


Figure 3.1: Axis definitions.

finite difference. This signal can be quite noisy, and will need appropriate filtering. In order to synchronize all signals with subscript 0, they all need to be filtered with this same filter.

Equation 3.1 can be turned into a control law using the matrix inverse or the pseudo-inverse:

$$\omega = \omega_0 + (G_1 + G_2)^{-1}(\nu - \dot{\Omega}_0 + G_2 L(\omega_c - \omega_0)), \tag{3.3}$$

but calculating the control input like this does not guarantee satisfying Equation 3.2. If the control inputs exceed the bounds, simply clipping them will result in different control moments than desired.

Instead, Equation 3.3 is replaced with a method that calculates the control inputs while respecting the limits and prioritization. This can be done with a weighted least squares (WLS) optimization. Since our system description (Equations 3.1 and 3.3) is in incremental form, we will first write it as a standard least squares problem through a change of variables:

$$v = Gu \tag{3.4}$$

subject to

$$u_{\text{min}} \leq u \leq u_{\text{max}}. \tag{3.5}$$

where the control objective is $v = \dot{\Omega} - \dot{\Omega}_0 + G_2 L(\omega - \omega_0)$, and the input is $u = \omega - \omega_0$. Note that because of the delay operator L, v depends on the value of u from the previous time step, which is already known and can be subtracted to obtain v for this time step. The limits u_{min} and u_{max} follow from these definitions and Equation 3.2.

3.3. Using the Active Set Method to Solve the Constrained Allocation Problem

Though in this chapter we will apply the algorithm to a quadrotor, for the control allocation we will also consider over-actuated systems. This means that we have to include a cost for actuator usage in the cost function, such that there is only one optimum. This will make the derived methodology easily applicable to other systems, like multirotors with more than four rotors, or some over-actuated hybrid systems like the Quadshot [60].

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

The sequential least squares problem is more difficult to solve than a least squares problem with a single cost function. This is why we adopt the WLS problem formulation from Härkegård [55], where the cost for errors in the control objective is combined with a cost for applying inputs:

$$\begin{aligned} C(u) = & \quad \|W_u\,(u-u_d)\|^2 + \gamma\,\|W_v\,(Gu-v)\|^2 \\ = & \quad \left\| \begin{pmatrix} \gamma^{\frac{1}{2}} W_v\,G \\ W_u \end{pmatrix} u - \begin{pmatrix} \gamma^{\frac{1}{2}} W_v\,v \\ W_u\,u_d \end{pmatrix} \right\|^2, \end{aligned} \tag{3.6}$$

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

$$A = \begin{bmatrix} \gamma^{\frac{1}{2}} W_v\,(G_1+G_2) \\ W_u \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} \gamma^{\frac{1}{2}} W_v v \\ W_u u_d \end{bmatrix}. \tag{3.7}$$

Now that the problem is formulated as a regular quadratic programming problem, it can be solved using the well known active set method [55, 61, 62], to find the inputs that minimize the cost function. The algorithm divides the inputs into a free set and an active set, which correspond to the inputs that are not saturated and to the actuators that are saturated respectively. The method disregards the inequality constraints for the free set, and for the active set W treats the constraints as equality constraints. At every iteration, it is evaluated if the division between active and free set is correct. For completeness, we explain our implementation of the active set method in Algorithm 1. Here, the matrix S contains the sign of active constraints on the diagonal.

The algorithm stops when the solution is optimal, or a maximum number of iterations is reached. Though the algorithm is guaranteed to find the optimum in a finite number of iterations, we may impose a maximum number of iterations that can be executed in a real time application. If the algorithm stops because the maximum number of iterations is reached, the solution will not be optimum. However, since the value of the cost function decreases at each iteration [62], the result will be better than at the start of the algorithm.

Algorithm 1: Active set method for WLS problem

```
Initialization:
$W = \{\emptyset\}, \quad u^0 = (u_{\max} - u_{\min})/2, \quad d = b - Au^0, S = [0]$
for $i = 0, 1, 2, .., n_{\max}$ do
    Determine the free columns in $A$:
        $A_f = A(:, h), \quad h \notin W$                                    (3.8)
    Determine the optimal perturbation by solving the following least
    squares problem for $p_f$:
        $d = A_f p_f$                                                        (3.9)
    Now $p$ is constructed from $p_f$ with zeros for the elements that are in $W$.
    if $u^i + p$ is feasible then
        $u^{i+1} = u^i + p$ and: $d = d - A_f p_f$
        The gradient and Lagrange multipliers are computed with:
            $\nabla = A^T d$ and: $\lambda = S\nabla$                        (3.10)
        if all $\lambda \geq 0$ then
            The solution $u^{i+1}$ is optimal $u = u^{i+1}$ ;
        else
            The constraint associated with the most negative $\lambda$ has to be
             removed from the active set $W$. Re-iterate with this active set.
    else
        The current solution violates a constraint which is not in $W$.
         Determine the maximum factor $\alpha$ such that $\alpha p$ is a feasible
         perturbation that activates constraint $j$, with $0 \leq \alpha < 1$. Update the
         residual $d$ and the solution $u^{i+1}$:
            $u^{i+1} = u^i + \alpha p$                                       (3.11)
            $d = d - A_f \alpha p_f$                                         (3.12)
         $u_j^{i+1}$ is now equal to $u_{\max}$ or $u_{\min}$. Finally, add $j$ to the active set
         and store the sign of the constraint: $S_{jj} = \text{sign}(p_j)$ with $j$ the index
         of the new active constraint.
```

3.3.1. Particularities for WLS applied to INDI

Since we are applying the WLS control allocation scheme to the INDI controller, the inputs are incremental. This means that the bounds on the input (increment) change every time step, and the solution for the increment at one time may not be feasible the next time step. The initial guess for the input, u^0, can therefore not be the solution of the previous time, as is often done for non-incremental controllers [54, 55, 61]. Instead, we take as initial input the mean of the maximum and minimum input increment:

$$u^0 = \frac{1}{2}(u_{\max} - u_{\min}). \tag{3.13}$$

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

3.3.2. Choice of Weighting Matrices

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

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

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

We choose $\gamma^{\frac{1}{2}} = 10000$ and for W_u we take the identity matrix, since all actuators are 'equal'. Do note that the relative scaling of the signals u and v plays a role here. Also note that, even though we give a lower weighting to some signals, they can still become dominant in the cost function if no bounds are applied. As an example, consider a quadrotor that has to climb five kilometers. In case of a simple PD controller without bounds, an enormous thrust will be commanded, leading to more cost in Equation 3.6.

3.3.3. Computational Complexity

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

An analysis of the performance of the active set algorithm on a benchmark problem set, with control objectives in $\mathbb{R}^3$ was done by Petersen and Bodson [61]. They report that the method is efficient in case of few actuators, but that it does not scale well with the problem size. Specifically, for 15 actuators or more, an interior point method is more efficient. Since our control objective is in $\mathbb{R}^4$, this point can be somewhere else.

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

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

3.4. Experiments

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

The hypothesis is that the WLS control allocation scheme, with the prioritization as defined in section 3.3.2, improves the tracking of pitch and roll when large yaw moments are required, as compared to calculating the inputs with the pseudo-inverse and clipping the result.

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

3.4.1. Experimental Setup

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

The INDI control algorithm of Chapter 2 is extended to also control the thrust in the body Z axis, next to the angular acceleration in three axes. This is done by adding a row to the matrices G_1 and G_2, containing the control effectiveness of the actuators on the thrust. A complete derivation will be presented in Chapter 4. Prior to the experiment, the control effectiveness matrices below were identified through test flights. The knowledge of the symmetrical geometry of the quadrotor was used to ease the estimation, and the coefficients of G_2 not belonging to the yaw axis were forced to zero.

$$G_1 = \begin{bmatrix} 18 & -18 & -18 & 18 \\ 11 & 11 & -11 & -11 \\ -0.7 & 0.7 & -0.7 & 0.7 \\ -0.4 & -0.4 & -0.4 & -0.4 \end{bmatrix} \cdot 10^{-3} \tag{3.14}$$

$$G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -65 & 65 & -65 & 65 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot 10^{-3} \tag{3.15}$$

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

3.4.2. Results

Figure 3.2 shows the Euler angles during the experiment for the pseudo-inverse on the left and for WLS on the right. 15 and 12 repetitions of the experiment are shown for the pseudo-inverse and WLS respectively. For WLS, two repetitions were rejected, because the pilot steered during the yaw step. The inputs to the actuators during the first repetition are shown in Figure 3.3.

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

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

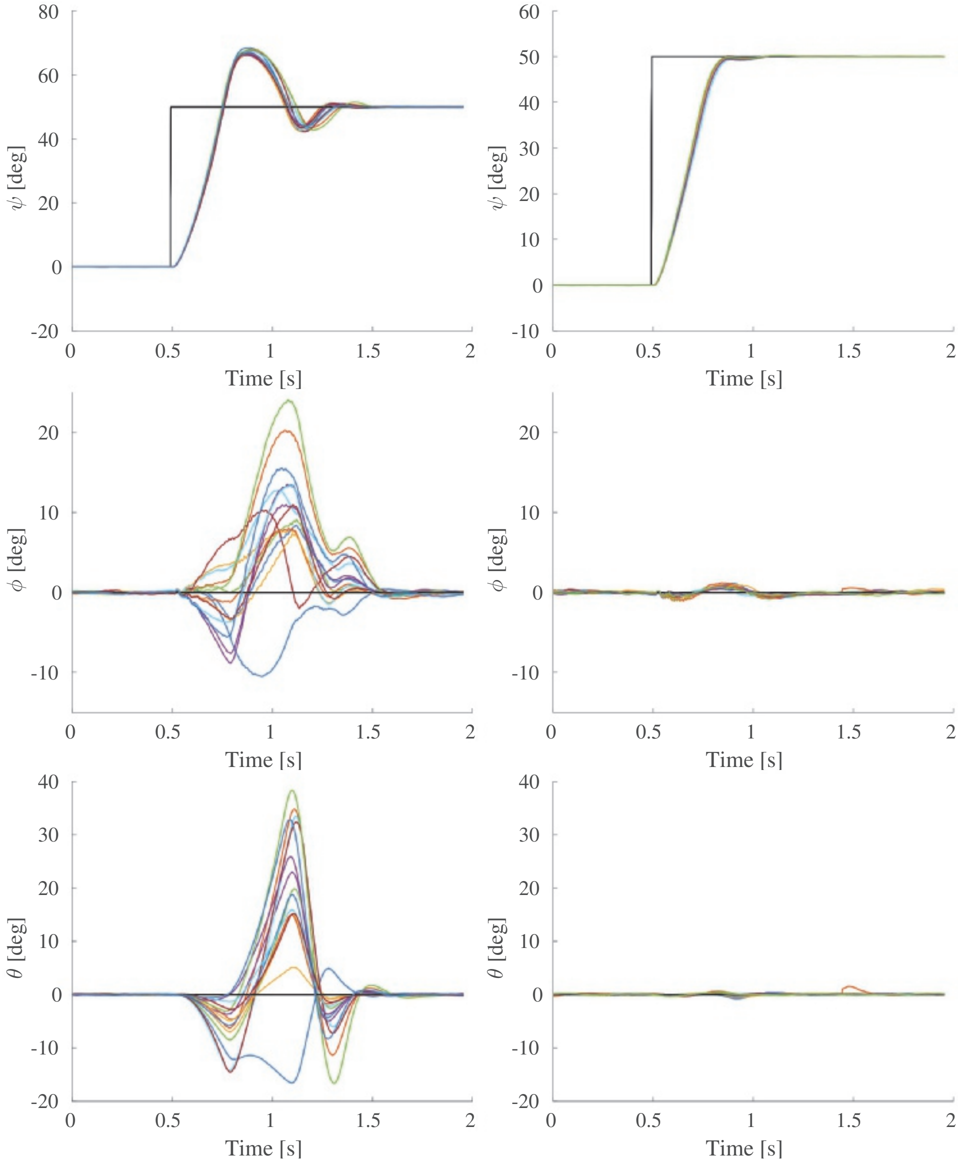


Figure 3.2: Euler angles during the experiment for the pseudo-inverse (left) and WLS (right).

However, the plots of pitch and roll show the merit of the WLS control allocation (note the different scale). To condense this information, we consider the maximum deviation of the roll and pitch angle from zero as a measure of the performance for each repetition. The mean and standard deviation of this maximum error per repetition is presented in Table 3.1.

Clearly there is a very significant improvement in the tracking of the pitch and

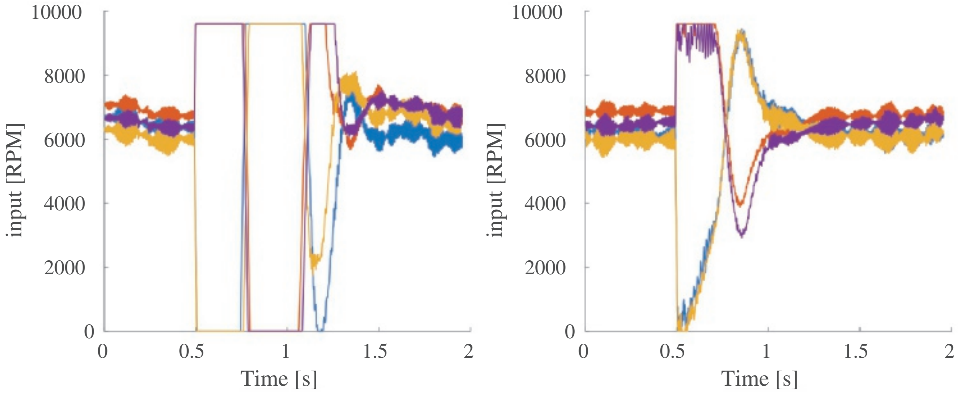


Figure 3.3: Inputs during the experiment for the pseudo-inverse (left) and WLS (right).

Table 3.1: Mean and standard deviation of the maximum pitch and roll error in degrees.

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

roll angles. We therefore conclude the hypothesis, that WLS improves the tracking of pitch and roll when producing large yaw moments, to be true.

Finally, from Figure 3.2 it does become apparent that there still is some small cross coupling between roll and pitch moments and the yaw moment for WLS. The exact cause is beyond the scope of this chapter, and may be a topic of future research, but there are possible explanations. For instance, the controller takes into account a linear control effectiveness, while this can be expected to be a quadratic one. Especially for large input changes, as is the case here, some error may be expected. Furthermore, we may consider the fact that for WLS, everything is combined into one cost function. This means that putting more weight on roll and pitch may reduce the error in tracking these angular accelerations, but will never bring it to zero. To improve this, the sequential formulation may be a solution.

3.5. Conclusion

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

The algorithm is readily applicable to other types of MAVs for which priorities in controlled axes can be defined, such as hexacopters, or even hybrid aircraft such as the *Cyclone* [63]. Future research will focus on how constraints in the guidance loop should be taken into account, and how this is affected by limits in the inner loop. Finally, given the strong disturbance rejection properties of the INDI controller, this control allocation scheme is expected to also increase the robustness against faults.

4

Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection

In the previous two chapters, an INDI controller was derived that can control the attitude of an MAV without an extensive model. It was shown that the controller can deal with disturbances and can manage priorities in the case of actuator saturation. In order to control the position of the MAV, an additional controller is needed.

In this chapter, the INDI controller that was developed to control angular accelerations, will be used to control linear acceleration as well. The controller is derived from first principles, and wind tunnel experiments confirm that the disturbance rejection properties are preserved in the outer loop. Furthermore, it is shown that the performance upholds outdoors with off-the-shelve GPS modules.

This chapter is based on the following article:
Smeur, E. J. J., de Croon, G. C. H. E., Chu, Q. P., Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection. Control Engineering Practice **73, 79 (2018).**

4.1. Introduction

Micro Aerial Vehicles (MAV) have the potential to perform many useful tasks, such as search and rescue [64], package delivery [65], aerial imaging [66], etc. In many of these applications, usage of autonomous MAVs can potentially result in significant cost reduction as compared to current practice. But in order to perform these tasks in an outdoor environment, the vehicles need to be able to control their position under the influence of wind gusts. This is especially true when flying close to obstacles, as a position error due to a wind gust might result in a collision.

Outdoor MAV missions can encounter significant gusts due to atmospheric turbulence [67]. Moreover, Orr *et al.* [68] showed that even a uniform wind can create a very non-uniform wind field in an urban environment. Computational fluid dynamics calculations showed that with a free stream velocity of 4.6 m/s, flow velocities ranging from 0 to 7.6 m/s are found around buildings. An MAV flying amidst these buildings can be expected to be subject to up to 7.6 m/s gusts. For most vehicles this can be considered a strong disturbance, although the effect depends on the weight and surface area of the vehicle. If such an MAV were to enter a building through an open window in case of a search and rescue mission, it would also experience a sudden change in wind speed. This scenario is especially challenging due to the confined space of a typical room. And even indoors, an MAV can be subject to aerodynamic disturbances, for instance caused by its own propeller backwash near walls [69].

Clearly, a controller that is able to counteract wind gust disturbances would be of great value. Currently widespread position control methods such as Proportional Integral Derivative control (PID), which are used even for aggressive control [70], do not perform well under the influence of gusts. PID gust rejection properties scale with magnitude of the gains, which is often limited by the GPS update frequency in outdoor scenarios. Moreover, the integrator term is generally slow in compensating persistent wind disturbances.

To cope with wind gusts, a solution could be to use onboard Pitot tubes to measure the relative velocity of the MAV with respect to the wind. The difference with the ground speed measured by a GPS module can provide an estimate of the local wind [71]. As disturbances may come from all directions, a minimum of six Pitot tubes would be necessary (two for each axis, as a Pitot tube can not measure negative airspeed). Alternatively, Mohamed *et al.* [72] used multiple multiprobe sensors to obtain flow pitch angle and velocity. Adding such an amount of extra sensors will increase the system complexity and cost. Furthermore, airspeed sensors are typically not reliable at low airspeeds.

Instead of using sensors, one could use a model of the MAV to estimate the wind velocity [73]. Waslander and Wang [74] used an extensive aerodynamic model to estimate wind velocities, with good results in simulation. The downside of this approach is that it requires a lot of parameters, which might even require wind tunnel tests as is done by Schiano *et al.* [75] and Tomić *et al.* [76]. If the model does not represent reality well enough due to modeling errors or airframe changes, the gust disturbance rejection performance will degrade.

In this chapter, a gust resistant controller is introduced through generaliza-

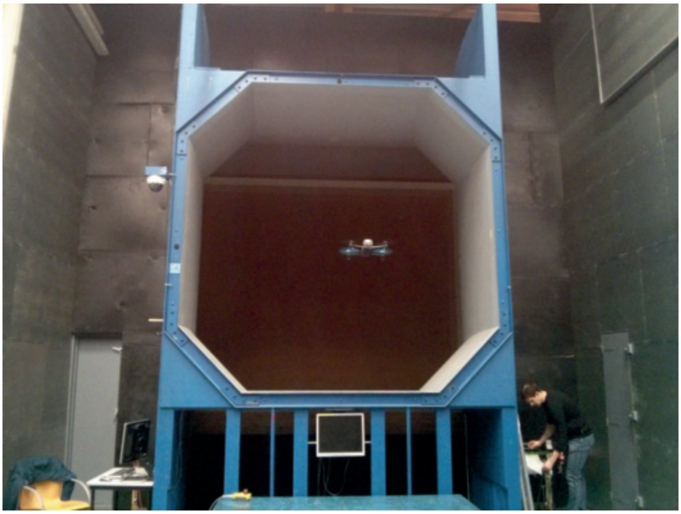


Figure 4.1: The quadrotor in front of the wind tunnel during one of the experiments.

tion of Incremental Nonlinear Dynamic Inversion (INDI) to the outer loop control. The idea is that both disturbances as well as control forces are measured by the accelerometer. This means that a desired acceleration can be achieved by incrementing the previous control input based on the difference between desired and measured acceleration. It is shown how to deal with filtering of noisy acceleration measurements, and how this integrates with the INDI attitude controller developed previously [52]. It is also demonstrated that the disturbance rejection capabilities of the INDI inner loop extend to the outer loop control.

The controller is implemented on a Parrot Bebop quadrotor running the Paparazzi open source autopilot software[1] [77, 78]. wind tunnel experiments show that the quadrotor can enter and leave the 10 m/s wind tunnel flow with only 21 cm maximum position deviation on average. A controller that uses a gain on the integrated error instead of the incremental controller, suffered 151 cm maximum position deviation upon entering and leaving the wind tunnel on average. To the best of the author's knowledge, this is the first time that a quadrotor is repeatedly flying in and out of a 10 m/s flow as part of a controller's disturbance rejection evaluation. A picture of the experiment is shown in Figure 4.1.

This chapter is an extension to the work presented at the Intelligent Robots and Systems conference [53]. Differences include: (1) the use of a large open jet wind tunnel as a more accurate and more powerful disturbance than the fan used previously. (2) Incorporation of the propeller thrust curve to calculate the total thrust of the drone. (3) An outdoors experiment based on GPS positioning, to demonstrate the performance in a realistic scenario. (4) The addition of a nonlinear method to calculate thrust vector increments.

[1] The INDI control method is incorporated in Paparazzi, allowing others to easily experiment with it.

4.1.1. Related work

Hoffmann *et al.* [79] developed an altitude controller that utilizes the vertical acceleration measurement. However, they fed the acceleration back, multiplied with a gain, without utilizing the physical relation between thrust and acceleration. In a different paper, they state that their PID position control implementation has little ability to reject disturbances from wind and translational velocity effects [80]. A vertical controller using the INDI principle was developed for a traditional helicopter in simulation by Simplicio *et al.* [46]. Only very limited sensor noise was taken into account, which did not require any filtering. Also, in both of these papers, by separating the vertical axis from the lateral axes, coupling can be expected. In this chapter it is shown that by inverting the control effectiveness for all axes, accelerations in each of these axes can be controlled.

Wang *et al.* [81] applied an acceleration feedback dynamic inversion approach to all axes of a quadrotor, and demonstrated accurate trajectory tracking capabilities. They mentioned robustness against disturbances, but did not analyze or demonstrate the controller response against disturbances. Also, the effects of accelerometer noise or filtering are not discussed. Additional differences with the work presented here are that with INDI there is no need for a reference model or command filtering, and that the approach of Wang is not incremental. This means that if a certain control input does not completely resolve a measured acceleration error (due to input modeling errors or uncertainties), the error will persist. In an incremental scheme the input can be incremented again to resolve angular acceleration errors.

4.2. Incremental nonlinear dynamic inversion for attitude control

An extended analysis of INDI for attitude control of MAVs is provided in previous work [52]. For completeness, an overview of the developed attitude controller, along with some new additions, is presented in this section. Consider the quadrotor shown in Figure 4.2. The distance from the center of gravity to each of the rotors along the X axis is given by l and along the Y axis by b.

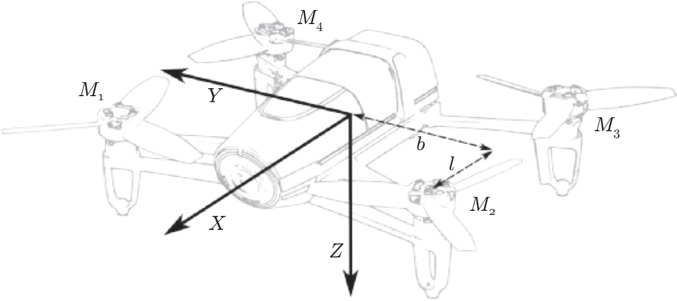


Figure 4.2: The Bebop Quadcopter used in the experiments with body axis definitions.

It is defined that $\mathbf{\Omega}$ is the angular rate vector of the vehicle and the angular rates of the propellers around the body Z axis are described with the vector $\boldsymbol{\omega}$. The thrust provided by all four rotors is denoted by T. Consider Eq. 4.1, which gives an expression for the angular acceleration and the thrust:

$$\begin{bmatrix} \dot{\mathbf{\Omega}} \\ T \end{bmatrix} = \mathbf{F}(\mathbf{\Omega}, \mathbf{v}) + \mathbf{G}(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}) \tag{4.1}$$

Here, $\mathbf{F}(\mathbf{\Omega}, \mathbf{v})$ is the function that describes the vehicle moments as a function of the angular rates and velocity. $\mathbf{G}(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}})$ is the function that maps the input and the derivative of the input to the angular acceleration and thrust. Note that the thrust force only depends on the rotational rate of the rotors, and so, the fourth row of the $\mathbf{F}(\mathbf{\Omega}, \mathbf{v})$ matrix is zero. Now a first order Taylor expansion can be applied:

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{\Omega}} \\ T \end{bmatrix} = \; & \mathbf{F}(\mathbf{\Omega}_0, \mathbf{v}_0) + \mathbf{G}(\boldsymbol{\omega}_0, \dot{\boldsymbol{\omega}}_0) \\ & + \frac{\partial}{\partial \mathbf{\Omega}}(\mathbf{F}(\mathbf{\Omega}, \mathbf{v}_0))|_{\mathbf{\Omega}=\mathbf{\Omega}_0}(\mathbf{\Omega} - \mathbf{\Omega}_0) \\ & + \frac{\partial}{\partial \mathbf{v}}(\mathbf{F}(\mathbf{\Omega}_0, \mathbf{v}))|_{\mathbf{v}=\mathbf{v}_0}(\mathbf{v} - \mathbf{v}_0) \\ & + \frac{\partial}{\partial \boldsymbol{\omega}}(\mathbf{G}(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}_0))|_{\boldsymbol{\omega}=\boldsymbol{\omega}_0}(\boldsymbol{\omega} - \boldsymbol{\omega}_0) \\ & + \frac{\partial}{\partial \dot{\boldsymbol{\omega}}}(\mathbf{G}(\boldsymbol{\omega}_0, \dot{\boldsymbol{\omega}}))|_{\dot{\boldsymbol{\omega}}=\dot{\boldsymbol{\omega}}_0}(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_0) \end{aligned} \tag{4.2}$$

First it can be recognized that the first two terms give the current angular acceleration and thrust: $\mathbf{F}(\mathbf{\Omega}_0, \mathbf{v}_0) + \mathbf{G}(\boldsymbol{\omega}_0, \dot{\boldsymbol{\omega}}_0) = [\dot{\mathbf{\Omega}}_0^T \quad T_0]^T$. The next two terms predict the change in moment due to changes airspeed and rotational rate. For these terms, both a detailed model and an airspeed estimate are needed. In absence of this information, these terms are omitted to simplify the equation and as such the terms are treated as a disturbance. This may lead to small errors in the angular acceleration prediciton. However, moments due to these terms eventually show up in the measured angular acceleration and are taken into account this way. Finally, it is assumed that over the operational domain, the partial derivatives of $\mathbf{G}(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}})$ do not change. Therefore, they can be approximated by the static matrices $\mathbf{G}_1$ and $\mathbf{G}_2$. These control effectiveness matrices are (4×4), because they contain the effectiveness of each of the four rotors on each of the axes roll, pitch, yaw and thrust.

$$\begin{bmatrix} \dot{\mathbf{\Omega}} - \dot{\mathbf{\Omega}}_0 \\ T - T_0 \end{bmatrix} = \mathbf{G}_1(\boldsymbol{\omega} - \boldsymbol{\omega}_0) + T_s\mathbf{G}_2(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_0) \tag{4.3}$$

Here, the sample time T_s is factored out of $\mathbf{G}_2$ to simplify future calculations. The angular acceleration $\dot{\mathbf{\Omega}}_0$ can be determined, by deriving it from the gyroscope using finite difference. This signal is often very noisy, because the rotating propellers lead to vibrations in the airframe. From Bacon *et al.* [34], the use of a second order filter is adopted, given by:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4.4}$$

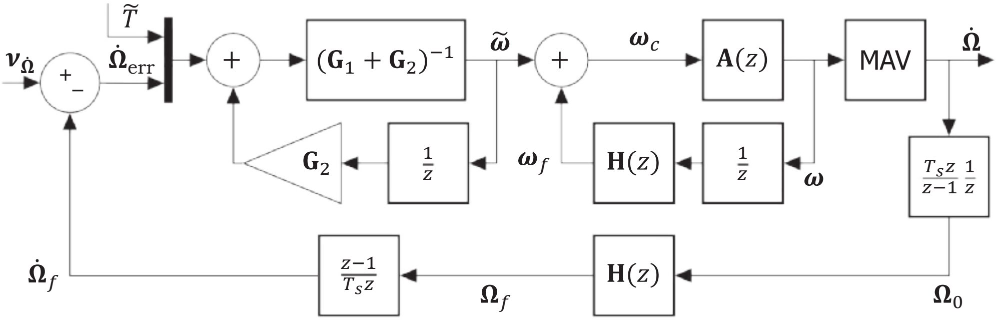


Figure 4.3: The inner INDI control structure.

This filter also introduces a delay. To be able to apply the Taylor expansion, terms with index 0 should be from the same moment in time. This is why all these terms should be filtered with the same filter, such that they are equally delayed. These terms are given the subscript f in Eq. 4.5.

$$\begin{bmatrix} \dot{\boldsymbol{\Omega}} - \dot{\boldsymbol{\Omega}}_f \\ T - T_f \end{bmatrix} = \mathbf{G}_1(\boldsymbol{\omega} - \boldsymbol{\omega}_f) + T_s\mathbf{G}_2(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_f) \tag{4.5}$$

By approximating $\dot{\boldsymbol{\omega}}$ and $\dot{\boldsymbol{\omega}}_f$ with finite difference, using the lag operator L as $\dot{\boldsymbol{\omega}} = (\boldsymbol{\omega}(k) - \boldsymbol{\omega}(k-1))/T_s = (\boldsymbol{\omega} - L\boldsymbol{\omega})/T_s$ and rearranging the terms, this equation becomes:

$$\begin{bmatrix} \dot{\boldsymbol{\Omega}} - \dot{\boldsymbol{\Omega}}_f \\ T - T_f \end{bmatrix} = (\mathbf{G}_1 + \mathbf{G}_2)(\boldsymbol{\omega} - \boldsymbol{\omega}_f) - \mathbf{G}_2 L(\boldsymbol{\omega} - \boldsymbol{\omega}_f) \tag{4.6}$$

This equation can now be inverted, to yield Eq. 4.7.

$$\boldsymbol{\omega}_c = \boldsymbol{\omega}_f + (\mathbf{G}_1 + \mathbf{G}_2)^{-1}\left(\begin{bmatrix} \boldsymbol{\nu}_{\dot{\Omega}} - \dot{\boldsymbol{\Omega}}_f \\ \tilde{T} \end{bmatrix} + \mathbf{G}_2 L(\boldsymbol{\omega}_c - \boldsymbol{\omega}_f)\right) \tag{4.7}$$

Here, $\boldsymbol{\nu}_{\dot{\Omega}}$ is the virtual control, which is the desired angular acceleration that has now become an input. In the next section it will be shown that the thrust increment $T - T_f$ is calculated in the outer loop, and is therefore denoted by $\tilde{T}$. The subscript c is added to $\boldsymbol{\omega}$ to indicate that this is the command sent to the motors.

The control diagram is shown in Figure 4.3. The input to this diagram is the angular acceleration virtual control and the output is the angular acceleration of the vehicle. The angular acceleration error, and the thrust increment go into the inversion of the control effectiveness. The increment in motor command is added to the feedback from the motors, which is filtered with the same filter as the angular acceleration. The actuator dynamics are denoted by $A(z)$.

4.2.1. Attitude control

The control law of Eq. 4.7 describes how to track angular accelerations. A PD controller can be used to provide the angular acceleration that will steer the vehicle towards a desired attitude. For the attitude feedback, a quaternion representation is used, as described by Fresk and Nikolakopoulos [37]. The calculation of the error quaternion $\mathbf{q}_{\text{err}}$ in terms of the reference $\mathbf{q}_{\text{ref}}$ and the state quaternion $\mathbf{q}_{\text{s}}$ is then:

$$\mathbf{q}_{\text{err}} = \mathbf{q}_{\text{ref}} \otimes \mathbf{q}_{\text{s}}^{*} \tag{4.8}$$

where $\otimes$ denotes the Hamilton product and * denotes conjugation. The reference angular rate is found using the vector part of the quaternion:

$$\boldsymbol{\Omega}_{\text{ref}} = K_{\eta} \begin{bmatrix} q_1^{\text{err}} & q_2^{\text{err}} & q_3^{\text{err}} \end{bmatrix}^T \tag{4.9}$$

The angular acceleration reference is then calculated from the rotational rate error, using a second gain K_{Ω}.

In order to find a theoretical basis for what the values of these gains should be, linear time-invariant systems theory will be used. That means that the attitude feedback needs to be simplified a bit, as is shown in Figure 4.4. In this figure small angles are assumed, in order to allow simple integration of the rates to obtain the attitude $\boldsymbol{\eta}$ in Euler angles.

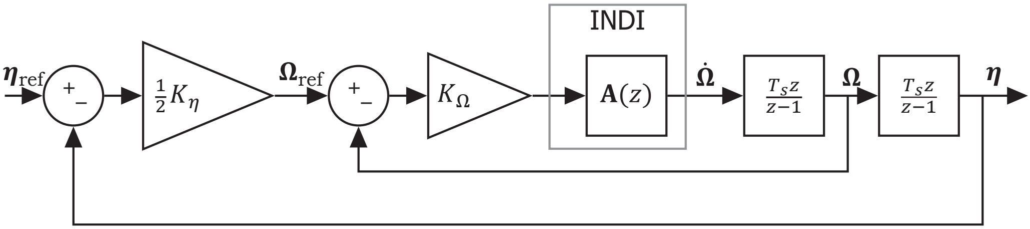


Figure 4.4: The design of the attitude controller for small angles, based on the closed loop response of the INDI controller. The feedback of the attitude is simplified for the sake of the analysis, and proper quaternion feedback is used on the real platform.

Because the proper quaternion integration is removed in Figure 4.4, there is a factor $\frac{1}{2}$ with K_{η}, since the quaternion derivative is defined as:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\Omega} \end{bmatrix} \tag{4.10}$$

In previous research [52] it was shown that if the assumptions, mentioned in the derivation of the controller, hold true, the transfer function from $\boldsymbol{\nu}_{\dot{\Omega}}$ to $\dot{\boldsymbol{\Omega}}$ is simply the actuator dynamics $\mathbf{A}(z)$. When the actuator dynamics can be modeled, for instance by first order dynamics, the P (K_{η}) and D (K_{Ω}) gains can be determined based on the desired poles and zeros of the system. For the Bebop, the actuator dynamics are modeled with first order dynamics as shown in Eq. 4.11, with $\alpha = 0.1$ at a sample frequency of 512 Hz.

$$A(z) = \frac{\alpha}{z - (1 - \alpha)} \tag{4.11}$$

Then the transfer function of the closed loop system from Figure 4.4 is as follows:

$$TF_{\eta_{\text{ref}}\to\eta} = \frac{\frac{1}{2}K_\Omega K_\eta \alpha T_s^2}{z^3-(3-\alpha)z^2+(3-2\alpha+K_\Omega\alpha T_s)z+(-1+\alpha-K_\Omega\alpha T_s+\frac{1}{2}K_\Omega K_\eta\alpha T_s)} \tag{4.12}$$

The gains can now be selected such that the poles are within the unit circle and the response is fast with little overshoot. With $K_\Omega = 28.0$ and $K_\eta = 10.7$, there is one pole at 0.964 and two complex poles at $0.968 \pm 0.0463i$.

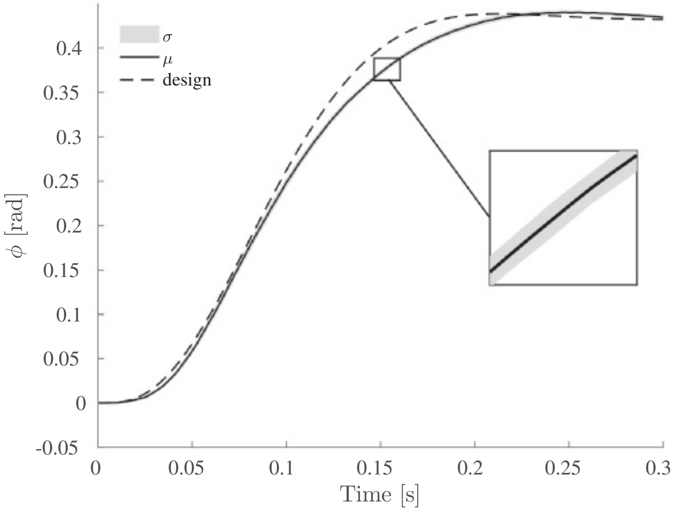


Figure 4.5: Comparison of the designed response and the actual response of the attitude of the quadrotor. The black line is the average, and the gray area one standard deviation of 25 repetitions.

An interesting question is how close the theoretically designed response of the attitude matches the actual attitude response of the quadrotor. To test this, both the above transfer function and the real quadrotor are subjected to a step input. For the real quadrotor, the step input is repeated 25 times, and the mean and standard deviation are shown in Figure 4.5. The response of the above transfer function is shown in the same figure in red. The difference between the designed response and the actual response is rather small: the error as a percentage of the step magnitude is maximum 6.4 % at 0.14 s. This means that this is a valid way of designing the P and D gains, based solely on the first order actuator dynamics model.

Some may argue that a nonlinear Lyapunov stability proof is more convincing than what is presented here. Note that, using a Lyapunov proof, the stability of a very similar quaternion feedback law was already proven by Tayebi and McGilvray [82]. Many recent papers on the topic of MAV control contain such a proof [83–86], and in each of these papers, an equivalent to the P and D gains exists. Further, in each of these papers, the claim is made, either explicit or implicit, that these gains can be chosen freely, irrespective of the Lyapunov proof. One might even get away with the feeling, reading these papers, that these gains have little influence on the final performance.

As anyone who has practical experience with control of drones can attest, these P and D gains are, in reality, not free to choose without consequences for the stability. There are most certainly non-negative, non-zero values of these gains for which the system is not stable, especially if the actuator dynamics are slow. This discrepancy exists because in these papers the actuator dynamics are always neglected. As is shown in this section, the actuator dynamics are crucial for the performance, as well as the stability. For different actuator dynamics, for example $\alpha = 0.02$, the transfer function of Equation 4.12 would even be unstable with the same gains.

Of course, it is still a good thing to be able to prove Lyapunov stability, even while neglecting actuator dynamics. It at least indicates that feedback is applied in the right direction, which may be troublesome in the case of feedback of Euler angles. However, that is not an answer to the complete stability question. Taking the actuator dynamics into account, for instance using the linear methods employed here, is an indispensable part of the stability and performance analysis.

4.2.2. Determining the control effectiveness

The control effectiveness can be derived from detailed knowledge of the system inertia and actuator forces. However, accurately measuring the vehicle inertia can prove to be difficult, time consuming and not very accurate. Alternatively, the control effectiveness matrices can be estimated using test flight data. A single test flight, with both the actuator inputs as well as the gyroscope and accelerometer data logged at high frequency, can provide enough information to determine the values for $\mathbf{G}_1$ and $\mathbf{G}_2$. Though some effort is required to make the vehicle fly stable without knowledge of the control effectiveness, this can be a very fast and efficient method to obtain the effectiveness. Further, the flight data will represent the system as it actually is, eliminating chances for modeling errors.

When the flight has been recorded, the control effectiveness can simply be calculated by finding the least squares solution of:

$$\begin{bmatrix} \ddot{\mathbf{\Omega}}_f \\ \dot{T}_f \end{bmatrix} = \begin{bmatrix} \mathbf{G}_1 & T_s\mathbf{G}_2 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\omega}}_f \\ \ddot{\boldsymbol{\omega}}_f \end{bmatrix} \tag{4.13}$$

where the derivatives can be calculated using the method of finite difference.

4.3. Incremental nonlinear dynamic inversion applied to linear accelerations

Now that the attitude of the quadrotor is controlled, it is possible to derive an incremental controller for the linear acceleration of the vehicle. Two reference frames will be used throughout this derivation; the body frame, as depicted in Figure 4.2, and the North East Down (NED) frame, which has its origin fixed to a point on the earth. Vectors in the body frame have a subscript B and vectors in the NED frame have subscript N. The subscripts will only be used to avoid confusion, the position ξ and velocity $\dot{\xi}$ of the quadrotor will always be in the NED frame.

The position dynamics are given by Newton's second law of motion:

$$\ddot{\xi} = \mathbf{g} + \frac{1}{m}\mathbf{F}(\dot{\xi}, \mathbf{w}) + \frac{1}{m}\mathbf{T}_N(\boldsymbol{\eta}, T) \tag{4.14}$$

Where $\ddot{\xi}$ is the acceleration of the MAV, $\mathbf{g}$ is the gravity vector and m is the mass. $\mathbf{F}$ is the aerodynamic force working on the airframe as a function of the velocity $\dot{\xi}$ of the MAV and the wind vector $\mathbf{w}$. $\mathbf{T}_N$ is the thrust vector in the NED frame as a function of the attitude $\boldsymbol{\eta} = [\phi, \theta, \psi]^T$ and the total thrust produced by the four rotors T.

The thrust vector in the NED frame can be obtained by taking the thrust vector in the body frame, given by $\mathbf{T}_B = [0, 0, T]^T$, and rotating it using the rotation matrix $\mathbf{M}_{NB}(\boldsymbol{\eta})$. Since the thrust vector in the body frame only has a Z component, only the last column of the rotation matrix is relevant. The thrust vector in the NED frame is therefore given by:

$$\mathbf{T}_N(\boldsymbol{\eta}, T) = \mathbf{M}_{NB}(\boldsymbol{\eta})\mathbf{T}_B = \begin{bmatrix} (\mathrm{s}\phi\mathrm{s}\psi + \mathrm{c}\phi\mathrm{c}\psi\mathrm{s}\theta)T \\ (\mathrm{c}\phi\mathrm{s}\psi\mathrm{s}\theta - \mathrm{c}\psi\mathrm{s}\phi)T \\ (\mathrm{c}\phi\mathrm{c}\theta)T \end{bmatrix} \tag{4.15}$$

where the sine and cosine functions are abbreviated by the letters s and c respectively. Now a first order Taylor expansion can be applied to Eq. 4.14, resulting in Eq. 4.16:

$$\begin{aligned} \ddot{\xi} = \quad & \mathbf{g} + \tfrac{1}{m}\mathbf{F}(\dot{\xi}_0, \mathbf{w}_0) + \tfrac{1}{m}\mathbf{T}_N(\boldsymbol{\eta}_0, T_0) \\ & + \tfrac{\partial}{\partial \dot{\xi}}\tfrac{1}{m}\mathbf{F}(\dot{\xi}, \mathbf{w}_0)|_{\dot{\xi}=\dot{\xi}_0}(\dot{\xi} - \dot{\xi}_0) \\ & + \tfrac{\partial}{\partial \mathbf{w}}\tfrac{1}{m}\mathbf{F}(\dot{\xi}_0, \mathbf{w})|_{\mathbf{w}=\mathbf{w}_0}(\mathbf{w} - \mathbf{w}_0) \\ & + \tfrac{\partial}{\partial \phi}\tfrac{1}{m}\mathbf{T}_N(\phi, \theta_0, \psi_0, T_0)|_{\phi=\phi_0}(\phi - \phi_0) \\ & + \tfrac{\partial}{\partial \theta}\tfrac{1}{m}\mathbf{T}_N(\phi_0, \theta, \psi_0, T_0)|_{\theta=\theta_0}(\theta - \theta_0) \\ & + \tfrac{\partial}{\partial \psi}\tfrac{1}{m}\mathbf{T}_N(\phi_0, \theta_0, \psi, T_0)|_{\psi=\psi_0}(\psi - \psi_0) \\ & + \tfrac{\partial}{\partial T}\tfrac{1}{m}\mathbf{T}_N(\phi_0, \theta_0, \psi_0, T)|_{T=T_0}(T - T_0) \end{aligned} \tag{4.16}$$

The first terms can be simplified to the acceleration at the previous timestep: $\mathbf{g} + \frac{1}{m}\mathbf{F}(\dot{\xi}_0) + \frac{1}{m}\mathbf{T}_N(\boldsymbol{\eta}_0, T_0) = \ddot{\xi}_0$. This acceleration can be obtained by rotating the specific force measured by the accelerometer in the body axes to the NED frame and adding the gravity vector. For the next two terms, the partial derivative of $\mathbf{F}$ with respect to $\dot{\xi}$ and $\mathbf{w}$, there is not a good estimate. For simplicity of the approach, the choice is made not to employ a model of the aerodynamic drag of the airframe. Moreover, it is very difficult, if not impossible, to predict how the wind is going to change. Therefore, the best guess for these terms is zero. Note that this does not mean that all aerodynamic forces are neglected. These forces will be measured with $\ddot{\xi}_0$. Finally, it is assumed that changes in ψ will be small, such that this term can be neglected. Combining this with Eq. 4.15 and 4.16 leads to:

$$\ddot{\xi} = \ddot{\xi}_0 + \frac{1}{m}G(\boldsymbol{\eta}_0, T_0)(\mathbf{u} - \mathbf{u}_0) \tag{4.17}$$

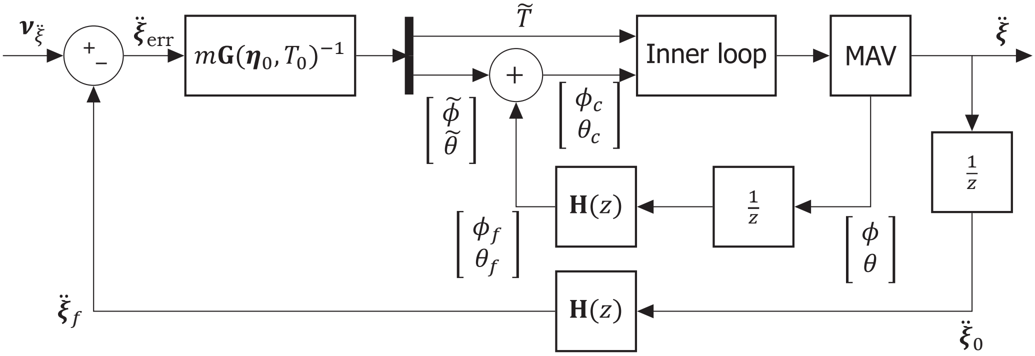

$\boldsymbol{\nu}_{\ddot{\xi}}$
+
−
$\ddot{\xi}_{\text{err}}$
$m\mathbf{G}(\boldsymbol{\eta}_0, T_0)^{-1}$
$\widetilde{T}$
+
$\begin{bmatrix}\widetilde{\phi} \\ \widetilde{\theta}\end{bmatrix}$
$\begin{bmatrix}\phi_c \\ \theta_c\end{bmatrix}$
Inner loop
MAV
$\ddot{\xi}$
$\mathbf{H}(z)$
$\frac{1}{z}$
$\frac{1}{z}$
$\begin{bmatrix}\phi_f \\ \theta_f\end{bmatrix}$
$\begin{bmatrix}\phi \\ \theta\end{bmatrix}$
$\ddot{\xi}_f$
$\mathbf{H}(z)$
$\ddot{\xi}_0$

Figure 4.6: The outer INDI control structure.

where $\mathbf{u} = \begin{bmatrix}\phi & \theta & T\end{bmatrix}^T$ and

$$\mathbf{G}(\boldsymbol{\eta}, T) = \begin{bmatrix} (\text{c}\phi\text{s}\psi - \text{s}\phi\text{c}\psi\text{s}\theta)T & (\text{c}\phi\text{c}\psi\text{c}\theta)T & \text{s}\phi\text{s}\psi + \text{c}\phi\text{c}\psi\text{s}\theta \\ (-\text{s}\phi\text{s}\psi\text{s}\theta - \text{c}\psi\text{c}\phi)T & (\text{c}\phi\text{s}\psi\text{c}\theta)T & \text{c}\phi\text{s}\psi\text{s}\theta - \text{c}\psi\text{s}\phi \\ -\text{c}\theta\text{s}\phi T & -\text{s}\theta\text{c}\phi T & \text{c}\phi\text{c}\theta \end{bmatrix} \tag{4.18}$$

The measured accelerations, necessary to obtain $\ddot{\xi}_0$, are typically noisy due to vibrations in the airframe introduced by the spinning propellers. Therefore, the accelerations need to be filtered. Like in the previous section, the delay of the filter needs to be accounted for. This is why also here, all terms with subscript 0 will be filtered with the same filter, and be given a subscript f. Then, if Eq. 4.17 is inverted, the INDI control law for linear accelerations is obtained:

$$\mathbf{u}_c = \mathbf{u}_f + mG^{-1}(\boldsymbol{\eta}_0, T_0)(\boldsymbol{\nu}_{\ddot{\xi}} - \ddot{\xi}_f) \tag{4.19}$$

$\ddot{\xi}$ is replaced with the virtual control $\boldsymbol{\nu}_{\ddot{\xi}}$ to indicate that this is now an input to the equation (the desired acceleration), and the subscript c is added to $\mathbf{u}$ to indicate that this is the command that will be sent to the inner loop controller. Also, the control increment is defined to be $\tilde{\mathbf{u}} = \mathbf{u}_c - \mathbf{u}_f$, so clearly Eq. 4.19 is an incremental control law.

Suppose the inner loop is filtered with filter f_1, and the outer loop with filter f_2. The thrust increment required by the inner loop is then $T - T_{f_1}$. The thrust increment calculated by the outer loop is $T - T_{f_2}$. It is only possible to pass on the thrust increment from the outer to the inner loop if these filters are equal. That is why for both loops, the filter described by Eq. 4.4 is used, with the same parameters.

4.4. Implementation

The implementation of the control law given by Eq. 4.19 is shown in Figure 4.6. The input of this diagram is the virtual control, and the output is the acceleration in NED frame. The acceleration in NED frame can be obtained from the accelerometer

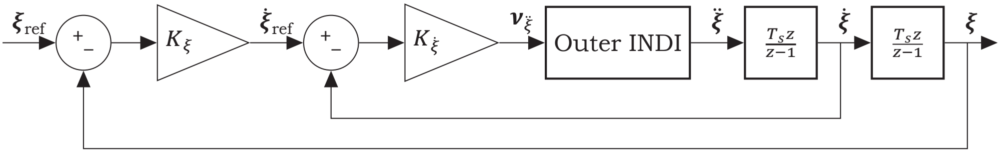


Figure 4.7: The PD controller for the position.

measurements with a simple rotation matrix and the addition of the gravity vector. Increments in roll, pitch and thrust are obtained from the error in acceleration multiplied with the inverse of the control effectiveness matrix. The roll and pitch increments are added to the filtered measurement of roll and pitch. Note how the increment in thrust command $\widetilde{T}$ goes directly into the inner loop.

4.4.1. Position control

In the previous sections, it was shown how the linear acceleration can be controlled using an INDI controller. To control the position of the MAV, an acceleration reference needs to be passed to the outer INDI controller that will steer the MAV towards its target position. This can be done by a Proportional Derivative (PD) controller, as is shown in Figure 4.7. The gains of this PD controller are manually tuned to give a fast response with little overshoot.

They depend mainly on two things: the update rate of the position estimate and the speed of the inner loop controller, which is only dependent on the actuator dynamics. This is the case because all other components are inverted in the inversion step of the inner and outer loop. Therefore, if these parameters are known in advance, one can come up with an estimate of the PD gains, for instance based on a pole/zero analysis.

4.4.2. Adaptive control effectiveness

In our previous work a Least Mean Squares algorithm was used to adapt the control effectiveness matrix of the rotors online [52]. Now, with respect to our previous work, a row is added to the control effectiveness matrix that predicts a change in thrust based on the actuator inputs. This row of the control effectiveness matrix can also be adapted online, together with the rest of the matrix. The LMS algorithm then becomes:

$$\mathbf{G}(k) = \mathbf{G}(k-1) - \boldsymbol{\mu}_2 \left(\mathbf{G}(k-1) \begin{bmatrix} \Delta\boldsymbol{\omega}_f \\ \Delta\dot{\boldsymbol{\omega}}_f \end{bmatrix} - \begin{bmatrix} \Delta\dot{\boldsymbol{\Omega}}_f \\ \Delta T \end{bmatrix} \right) \begin{bmatrix} \Delta\boldsymbol{\omega}_f \\ \Delta\dot{\boldsymbol{\omega}}_f \end{bmatrix}^T \boldsymbol{\mu}_1 \tag{4.20}$$

where

$$\mathbf{G}(k) = \begin{bmatrix} \mathbf{G}_1(k) & \mathbf{G}_2(k) \end{bmatrix} \tag{4.21}$$

This means that the effectiveness of the motors with respect to the thrust can also be adapted online. This can be important if the weight of the vehicle changes during flight, for instance when dropping a payload. Given a flight with enough

Table 4.1: Adaptation of the control effectiveness in the thrust axis in 10^{-3} m/s^2/rpm

	M_1	M_2	M_3	M_4
Adaptation from -0.35	-0.77	-0.73	-0.58	-0.63
Standard deviation	0.025	0.024	0.012	0.016
Adaptation from -1.13	-0.76	-0.73	-0.59	-0.65
Standard deviation	0.021	0.033	0.030	0.027
Offline	-0.76	-0.72	-0.57	-0.63

excitation of the control input and limited disturbances, the control effectiveness converges to the control effectiveness calculated offline.

To demonstrate this, 10 flights were performed with the adaptive estimation enabled. The flights were manually piloted, with constant maneuvering to ensure enough excitation. Five of these started with the effectiveness of each rotor on the thrust set equal to -0.35, and five started with the effectiveness equal to -1.13. From these flights, the adapted control effectiveness was recorded 30 seconds after takeoff. Finally, one flight of two minutes was performed, and the logged flight data was processed offline with the method of Section 4.2.2 as a comparison.

The control effectiveness values are shown in Table 4.1, averaged for the cases with five flights. Clearly, after 30 seconds the control effectiveness of each motor on average has converged very close to the value calculated offline. Additionally, it can be observed that the identified control effectiveness values differ quite a bit between motors. These differences are naturally observed from the test-flight driven identification, whereas otherwise these details would be difficult to obtain.

In the next section, Figure 4.8 shows the average thrust curve of the actuators. One possible explanation of the differences observed in Table 4.1, is that the motors are operating at different RPM. However, the average RPM of the two minute flight was 126, 126, 125 and 123 for motors one through four respectively.

4.4.3. Estimation of the thrust

Throughout the derivation of the outer loop INDI controller, use is made of the thrust T, for instance in the matrix $G(\boldsymbol{\eta}, T)$. One possibility would be to measure the specific force in the body Z axis with the accelerometer, and use this as an estimate for the specific thrust ($\frac{T}{m}$). This approach works well while hovering, but can lead to errors when there are other (aerodynamic) forces in the body Z axis. These forces occur for instance at high speed steady flight, when the drone has a high bank angle.

Therefore, static thrust measurements are used to model the thrust/rotational rate curve of the propellers. The quadrotor was mounted upside down on a scale to obtain a measurement of the produced thrust. The rotational rate of the propellers was obtained from the internal rpm measurement. The resulting average thrust measurement per propeller as function of the rotational rate is shown in Figure 4.8.

A quadratic function showed a good fit with the data. This function is used for the thrust estimate in the calculations of the controller.

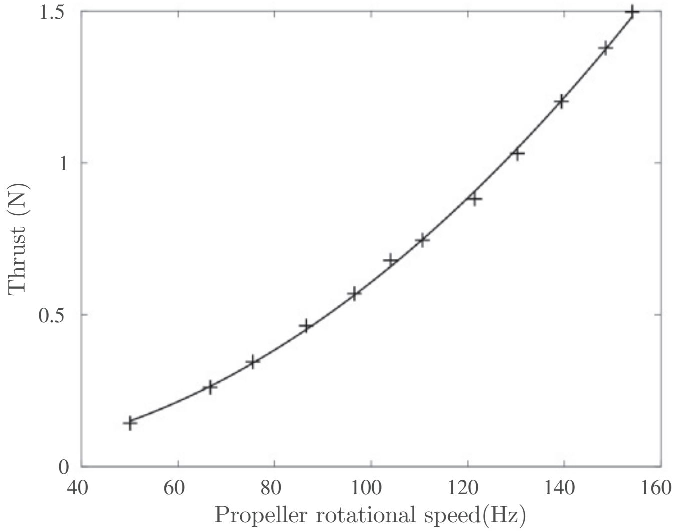


Figure 4.8: Thrust measurements for different rotor speeds along with a second order approximation.

Of course, the actual thrust of the propellers may be slightly different in a real flight, due to a different inflow. Furthermore, since the propellers have a quadratic thrust curve, their control effectiveness changes depending on their current rotational rate. In this chapter, it is assumed that the control effectiveness of the rotors with respect to the specific force can be approximated by a static one. This removes the need to recalculate $(\mathbf{G}_1 + \mathbf{G}_2)^{-1}$ after every time step, enhancing the speed of the algorithm.

Errors introduced by these simplifications are expected to have low impact, because of the incremental nature of the controller. If an increment in thrust does not lead to the desired acceleration, another increment is applied. This way, small errors in the control effectiveness are handled naturally.

4.4.4. Linearization

In the previous section, a first order Taylor expansion was used to derive the INDI control law. However, from Equation 4.15 it can be seen that the force is actually very nonlinear in terms of roll and pitch. In Equation 4.18 it can be seen that some of the derivatives can even change sign, for instance $\frac{\partial \ddot{z}}{\partial \phi}$ for different values of ϕ.

What this means in practice is that if the increments in the input are large, because suddenly a large lateral acceleration is required, they will result in a different acceleration than intended. This will be measured by the accelerometer, and subsequent increments in the inputs will eventually lead to the right acceleration. But it might be more effective to implement a nonlinear method of finding increments in the input that give exactly the desired increment in the acceleration.

Since the nonlinear function of the inputs is available, it is possible to do a nonlinear inversion. At the same time, it is preferable to keep the incremental structure, because there is no accurate estimate of the aerodynamic forces $\mathbf{F}$. Referring to Eq. 4.14, it is possible to subtract the same formula a short time instant earlier:

$$\begin{aligned} \ddot{\boldsymbol{\xi}} - \ddot{\boldsymbol{\xi}}_0 = \; & \mathbf{g} - \mathbf{g}_0 + \tfrac{1}{m}\mathbf{F}(\dot{\boldsymbol{\xi}}, \mathbf{w}) - \tfrac{1}{m}\mathbf{F}(\dot{\boldsymbol{\xi}}_0, \mathbf{w}_0) \\ & + \tfrac{1}{m}\mathbf{T}_N(\boldsymbol{\eta}, T) - \tfrac{1}{m}\mathbf{T}_N(\boldsymbol{\eta}_0, T_0) \end{aligned} \tag{4.22}$$

It is assumed that changes in gravity and the aerodynamic forces are small during this small time instant:

$$\ddot{\boldsymbol{\xi}} - \ddot{\boldsymbol{\xi}}_0 = \frac{1}{m}\mathbf{T}_N(\boldsymbol{\eta}, T) - \frac{1}{m}\mathbf{T}_N(\boldsymbol{\eta}_0, T_0) \tag{4.23}$$

This equation relates an increment in thrust vector to an increment in acceleration. The current thrust vector can be calculated based on the attitude and rotational rate of the rotors. This gives an expression for the new thrust vector:

$$\mathbf{T}_N(\boldsymbol{\eta}, T) = m(\ddot{\boldsymbol{\xi}} - \ddot{\boldsymbol{\xi}}_0) + \mathbf{T}_N(\boldsymbol{\eta}_0, T_0) \tag{4.24}$$

How the thrust vector depends on the thrust and attitude is described by Eq. 4.15. A nonlinear inversion of this equation provides expressions for the thrust, roll and pitch commands:

$$T = ||\mathbf{T}_N|| \tag{4.25}$$

$$\phi_c = \arcsin\left(\frac{\sin(\psi)T_{N_x} - \cos(\psi)T_{N_y}}{T}\right) \tag{4.26}$$

$$\theta_c = \arcsin\left(\frac{\cos(\psi)T_{N_x} + \sin(\psi)T_{N_y}}{T\cos\phi_c}\right) \tag{4.27}$$

This allows us to find a new attitude and thrust that will satisfy a desired acceleration, without linearizing the input function. At the same time, the incremental structure is retained, as the new thrust vector is calculated based on the previous one. For the nonlinear case, the same argument holds regarding the filtering as for the linearized case: if the acceleration is filtered, the other signals with subscript 0 should also be filtered. This is shown below:

$$\mathbf{T}_N(\boldsymbol{\eta}, T) = m(\ddot{\boldsymbol{\xi}} - \ddot{\boldsymbol{\xi}}_f) + \mathbf{T}_N(\boldsymbol{\eta}_f, T_f) \tag{4.28}$$

4.4.5. Filtering

Both the measured accelerations as well as the rates are filtered to remove noise. In the derivation of the INDI controller, it was shown that these signals should be filtered with the same filter. This way, the delay in both loops is synchronized and the thrust increment can go from the outer loop to the inner loop.

Previously, it was shown that the filter choice has an effect on the disturbance rejection [52]. For the attitude loop, the response to a disturbance is given by

$(1 - A(z)H(z)z^{-1})$. By taking a filter with a higher cutoff frequency, and therefore less delay, disturbances will be rejected faster. On the other hand, more noise will end up in the control signals. Since the inner and outer INDI loops are connected and need to use the same filter, this trade-off should be considered for both loops simultaneously. For the experiment, a filter is chosen with a ω_n = 50 rad/s and ζ = 0.55.

4.4.6. Accelerometer bias

The outer loop INDI controller is somewhat sensitive to accelerometer biases. Because the accelerometer measurement is fed back to control the acceleration, an offset in the measurement will result in an offset in the actual acceleration as well. This means that the quadrotor will maintain its position not at a position with zero error, but at a position where the position error times the P gain gives a required acceleration equal to minus the acceleration offset.

This problem does not arise in the inner loop, where the angular acceleration is bias free. This is because in calculating the derivative of the rates, the bias disappears from the signal. This is not the case for the outer loop, so it is necessary to estimate the accelerometer bias in order to remove it. As a second measurement, the velocity and position from GPS or an indoor positioning system can be used. The velocity measurement can be derived to obtain a bias free acceleration measurement. Because of the low update rate, this signal is not really viable for feedback.

However, the acceleration obtained from velocity can be used to determine the accelerometer bias. The derived acceleration has to be rotated to the IMU axes in order to take the difference with the acceleration measured by the IMU. The accelerometer bias is a signal that is assumed to vary only very slowly. This is why the acceleration difference is filtered with a second order filter with a natural frequency of 0.25 rad/s. This removes all noise while keeping the important bias information. This is only a very simple method of finding the accelerometer bias, it could alternatively be done with a Kalman filter.

4.5. Wind tunnel experiment

The INDI outer loop controller, described in the previous sections, is designed to perform well against disturbances, such as wind gusts. If this is actually true in in the real world, should be tested in a repeatable, controlled experiment. It has to be known exactly what the disturbance is, to be able to assess the disturbance rejection performance. That is why the experiment is performed indoors, where the disturbance can be precisely determined and the position of the drone can be accurately measured.

4.5.1. Test setup

The source of the wind disturbance is the Open Jet Facility of the TU Delft Aerospace department. It has a 2.85 m by 2.85 m cross section, and is capable of velocities up to 30 m/s. A picture of the drone flying in front of the wind tunnel is shown in Figure

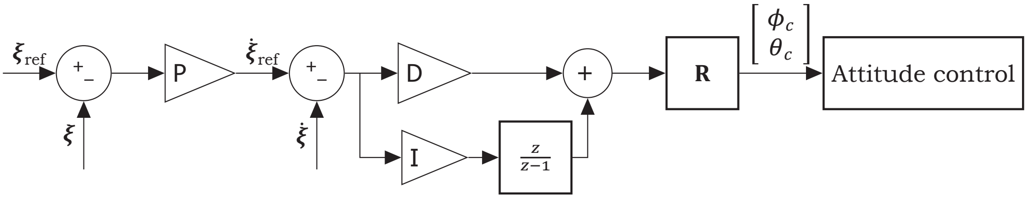


Figure 4.9: The horizontal PID controller used for comparison.

4.1. Because the wind tunnel is not capable of rapidly increasing or decreasing its velocity, the quadrotor has to fly in and out of the wind tunnel exhaust to simulate a gust. This is done by letting the quadrotor alternate between two waypoints every 14 seconds, one being in the center of the of the wind tunnel exhaust, and one being outside the wind tunnel flow, at two meters distance. For the experiment, the wind tunnel is set to 10 m/s, while the maximum speed of the Bebop is 13 m/s, according to the manufacturer. Needless to say, flying in and out of this flow is a simulation of a very strong gust.

Though this is the first time such an experiment is ever performed in a wind tunnel, it bears some resemblance to an experiment performed earlier, using a big fan [53]. However, the maximum airspeed measured 1 m in front of the fan was only 4.0 m/s, a relatively small disturbance. Moreover, the flow was not very homogeneous, as the airspeed was only 1.3 m/s at the centre of the fan. The use of a professional wind tunnel makes the results much more quantitative.

The INDI loop basically controls the acceleration of the MAV, whith a PD controller providing the acceleration reference. From this perspective, it is a variation of a PID controller, where the INDI loop replaces the integral of the PID controller. This is why the performance of INDI will be compared with a regular PID controller. The PID controller is manually tuned to give the fastest response possible to a 5m step input, without oscillation. Energy efficiency is not considered in the tuning of the controller. Both the INDI as well as the PID outer loop controllers make use of the inner loop INDI controller for attitude control. For the PID controller, the P, I and D gains work directly on the position and velocity to produce a reference roll, pitch and thrust, as is shown in Figure 4.9. Here **R** is a matrix defined by:

$$\mathbf{R} = \begin{bmatrix} -\sin\psi & \cos\psi \\ -\cos\psi & -\sin\psi \end{bmatrix} \tag{4.29}$$

In tuning the PID gains, there is a trade-off to be made. By increasing the integral gain, faster offset compensation can be obtained. This way the quadrotor can adjust to the disturbance of the wind tunnel faster. However, with a high integral gain, the quadrotor will also have more overshoot in reference tracking tasks such as sudden position changes. This trade-off is non-existent for the INDI controller. Table 4.2 and 4.3 present the parameters that have been used for the INDI and PID controllers respectively.

It is possible to do a crude comparison between the K_ξ and $K_{\dot{\xi}}$ gains of the INDI controller and the P and D gains of the PID controller. Around hover, the

Table 4.2: INDI parameters.

Parameter	Value	Unit
ω_n	50	rad/s
ζ	0.55	
K_Ω	10.7	(rad/s^2)/(rad/s)
K_η	28.0	
K_ξ	0.7	(m/s)/m
$K_{\dot{\xi}}$	1.5	(m/s^2)/(m/s)

Table 4.3: PID parameters.

Parameter	Value	Unit
P	0.65	(m/s)/m
I	0.11	rad/(m/s)/s
D	0.2	rad/(m/s)

virtual control is related to the change in commanded attitude angle through a division by gravity, assuming small angles. This means that $K_{\dot{\xi}}$ should be divided by 9.81, so K_ξ and $K_{\dot{\xi}}$ become 0.70 (m/s)/m and 0.15 rad/(m/s) respectively. For the PID controller, the corresponding P and D gains are 0.65 (m/s)/m and 0.20 rad/(m/s) respectively. Though these gains are not exactly the same, the goal of this crude comparison is to show that both controllers have roughly the same gains. Since the disturbance rejection properties will be considered, the integral gain will play the biggest role.

The MAV used for the experiments is the Bebop quadrotor from Parrot. Instead of the stock firmware, it is running the Paparazzi open source autopilot system. An infrared motion tracking system called 'Optitrack' was used to obtain position information. This system can measure the drone's position with millimeter accuracy at a frequency up to 120 Hz. But because the experiment should be realistic for outside scenarios and since most Global Positioning System (GPS) modules can only provide position updates at 4 Hz, the data was only sent to the drone at a frequency of 4 Hz. The control algorithm, as well as the onboard accelerometer and gyroscope, were running at 512 Hz. In an outdoor scenario, millimeter accuracy might not be achievable with off the shelve GPS modules. But even though the position might be off in such a case, gusts will still be rejected the same way as in this indoor experiment, as the INDI controller is based on the accelerometer.

4.5.2. Results

First, consider Figure 4.10. It shows the acceleration in the north axis, which is the axis in which the wind tunnel is blowing. The acceleration is filtered with a second order filter with $\omega_n = 20$ rad/s and $\zeta = 0.7$. The quadrotor starts besides the wind tunnel, and at 0.0 seconds, the quadrotor is commanded to fly to the waypoint in front of the wind tunnel. The moment the quadrotor flies into the wind stream is clearly visible in the figure due to the large acceleration spike, deviating from the reference acceleration. Due to this acceleration error, the INDI controller will increment the control inputs in order to make the acceleration track the reference again. About half a second after the start of the disturbance, the acceleration coincides with the reference acceleration, effectively having counteracted the disturbance. At that point, the quadcopter has built up a speed and position error in the north axis. The quadrotor needs a positive acceleration after the disturbance to bring these errors back to zero.

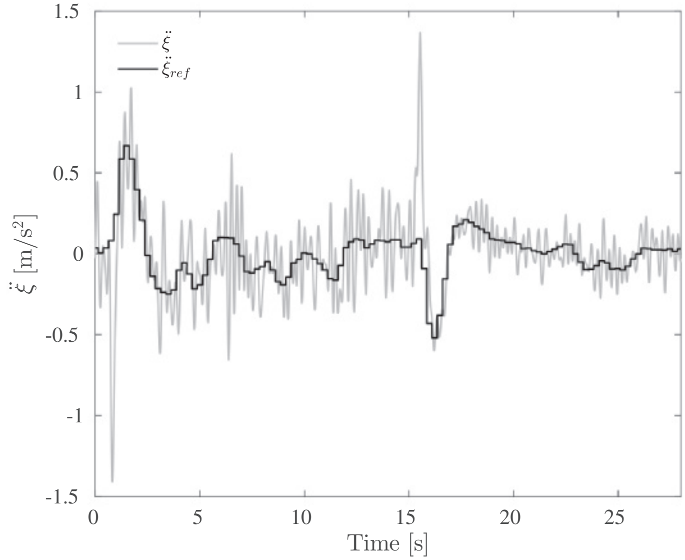


Figure 4.10: Acceleration in the north direction for the INDI controller.

The same thing happens at 15.4 seconds, when the quadrotor is commanded to fly out of the wind again. Now the sudden absence of wind results in a disturbance in the opposite direction. What also can be observed from this figure is that the accelerometer measures a more high frequency signal when flying in the wind. This could be due to the airflow containing some turbulence.

Figure 4.11 shows the position along the X_N axis, for both the INDI and PID controllers. The figure shows the average of seven times the same maneuver, along with one standard deviation. For INDI, it can be observed that a position error of 0.21 m occurs upon entering (2.0 seconds) and 0.20 m upon leaving (16.6 seconds) the wind tunnel. This position error is counteracted within three seconds after it occurred.

Compare this with the position for the PID controller in the same figure. The maximum error is 1.51 m, and it takes longer for the position error to be counteracted as compared to the INDI controller. One thing to note is that, when the vehicle is flying in front of the wind tunnel and there are no changing disturbances, the PID controller shows less variance between flights than the INDI controller.

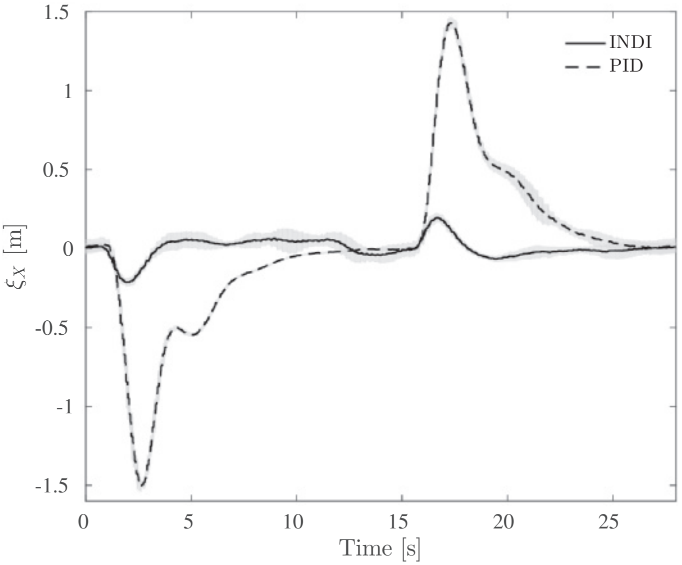


Figure 4.11: Position in the North direction for the INDI and PID experiment. The lines are the average of seven repetitions, and the shaded areas indicate one standard deviation.

This difference may be attributed to the fact that the INDI controller is using the accelerometer for feedback. Though the accelerometer measurement is filtered, it is still a bit noisy. A filter with more high frequency attenuation could have been used, but this would make the disturbance rejection of the controller slightly slower, because such a filter has more delay.

A top view of the experiment is shown in Figure 4.12. From this figure the difference in performance becomes apparent. The figure shows the entire flight, from takeoff until landing. For the PID controller, one can see how it is blown in the negative X_N direction upon takeoff, entering the wind tunnel, and how it overshoots in a straight line upon landing, leaving the wind tunnel flow. The INDI controller is able to cope much better with the sudden wind changes during taking off and landing.

4.6. Outdoor takeoff with wind

The experiment in the wind tunnel is great from a scientific point of view, as it allows us to compare different controllers subject to exactly the same disturbance. On the other hand, since an Optitrack system was used for position estimation, it might not be clear if the controller can provide the same performance in an outdoor scenario. That is why a second experiment is performed; outdoors and with a standard off-the-shelve GPS receiver.

One of the situations in which an MAV needs to cope with a sudden wind disturbance, is during takeoff on a windy day. When on the ground, the ground is compensating the drag from the wind. But when the drone takes off, the wind

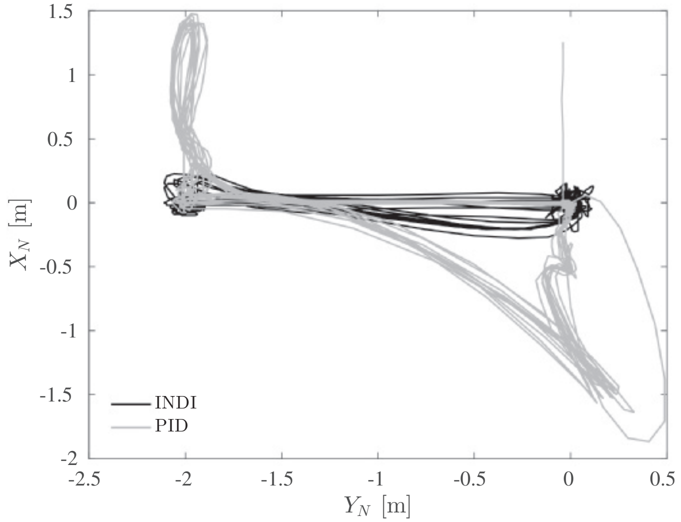


Figure 4.12: Top view of the experiment for the PID and INDI controllers. The wind tunnel is blowing in the negative X_N direction, and is located at $-1.425 < Y_N < 1.425$.

force will accelerate the drone. Therefore, a control action is needed to counteract the wind and maintain position.

4.6.1. Test setup

Since the acceleration is measured with the accelerometer, it is expected that the INDI controller will compensate for the wind force very fast. A PID controller that does not use this information, on the other hand, is expected to drift a bit, until it has gained some error in position and velocity that causes it to steer back. The integrator part will remove the steady state error over time.

Like before, the first version of the Bebop quadrotor is used for this experiment. As opposed to the second version of the Bebop, the first version that is used for this experiment has a low quality GPS. With the built-in GPS, the disturbance rejection performance is hard to evaluate, as the position estimate will move around quite a bit, regardless if the drone is moving or not. This is why the quadrotor is equipped with an external Ublox Neo M8N through a USB connection. This GPS module is commercially available, and the second version of the Bebop even ships with this module built in.

Like with the wind tunnel experiment, the case with outer loop INDI will be compared with a PID outer loop controller. The PID controller has the same gains as in the wind tunnel experiment, just like the P and D gains that produce the acceleration reference for the INDI controller are the same.

4.6.2. Results

On the day of the outdoor takeoff experiment an average wind speed of 5.1 m/s was reported by the Dutch Meteorological Institute (KNMI). Over the course of one

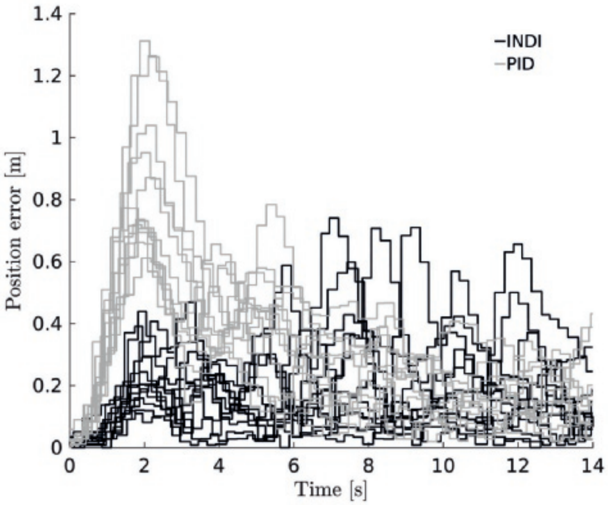


Figure 4.13: The horizontal position error during the outdoor takeoff experiment.

and a half hour, first twelve flights were performed with the PID controller, and then thirteen with the INDI controller. The flights were performed one after the other, without breaks. It is assumed that on average, the wind during the INDI flights was the same as during the PID flights, even though a fluctuation of the wind speed between flights was observed.

One of the flights with the INDI controller was rejected, as from the data it became clear that the state estimation filter had not converged prior to takeoff. The state estimation error leads to a bias in the NED acceleration, which in turn leads to a position offset, as discussed above.

The position error can be seen in Figure 4.13. The average position error is shown in Figure 4.14.

The position reference was reset to the current position just before each flight, so all flights start with a position error close to zero. As expected, during the takeoff INDI performs much better than the PID controller. It can be seen from Figure 4.14 that the INDI controller produces on average a maximum position error of 0.24 m as compared to 0.85 m for the PID controller.

Though Figure 4.14 shows that the average error after some time is the same for both controllers, it appears from Figure 4.13 that there are some runs for the INDI controller with relatively large errors. These errors are especially large if they are compared to the position error that is the result of the takeoff in the wind, which was expected to be the main disturbance. Closer inspection of some of these datasets show that when these errors occur, the acceleration measured by the accelerometer does not correspond with the position and velocity measured by the GPS. This may indicate that these errors are caused by GPS errors, perhaps upon changing between satellites. This could perhaps be solved with a better state

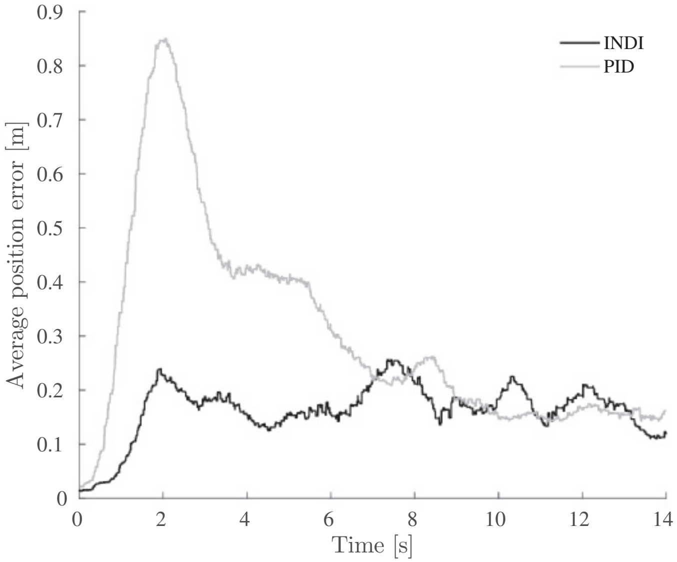


Figure 4.14: The average horizontal position error during the outdoor takeoff experiment.

estimation algorithm, but that is beyond the scope of this research.

4.7. Nonlinear increment

As described in the section Implementation, the increment in thrust vector $\mathbf{T}_N(\boldsymbol{\eta}, T)$ can also be computed without linearizing. The linearization will give a small error if the virtual control $\boldsymbol{\nu}_{\ddot{\xi}}$ is small. But for large values of $\boldsymbol{\nu}_{\ddot{\xi}}$, the error will be more significant.

It might be the case that, while such an incorrect increment in thrust vector is being executed and the quadrotor is rotating, a difference with the expected acceleration is already measured, and subsequent increments correct the thrust increment such that it will give the desired acceleration. It will depend for a large part on the cutoff frequency of the measurement filter if this will be fast enough. If the cutoff frequency of the filter is low, the delay may make the rejection of this disturbance too slow.

4.7.1. Test setup

To assess whether the linearization is accurate enough for large acceleration changes, an experiment is devised. In this experiment, from a hover initial condition, the quadrotor is commanded an acceleration of (0,4,0) m/s^2 for half a second, and then (0,-4,0) m/s^2 for another half second. This way, it will go from accelerating in one direction, to accelerating in the other direction, resulting in a large acceleration change. The maneuver takes place in the Y-axis, but considerable response differences in the Z-axis are expected.

This flight plan will be used for the Bebop quadrotor controlled with linearized

INDI and nonlinear INDI. For the linearized version the hypothesis is that the quadrotor will suddenly give very little thrust when the large change in acceleration is commanded, because of the derivative of the vertical component of $\mathbf{T}_N(\boldsymbol{\eta}, T)$ with respect to thrust. Due to this sudden decrease in thrust, the quadrotor is expected to slightly descend, before the vertical acceleration is measured and the thrust is increased again. For the nonlinear version the hypothesis is that the thrust command will remain roughly the same, and little change in altitude is expected.

The experiment is performed in the indoor flight arena facility at the faculty of Aerospace Engineering in Delft. The quadrotor is hovering based on the position feedback it receives from the tracking system. It does not use this position information during the maneuvers, because the acceleration reference during these maneuvers is predefined. The control effectiveness matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ are determined prior to the experiment using the adaptive algorithm. The experiment is repeated 25 times for both conditions.

4.7.2. Results

Figure 4.15 shows the acceleration in the Z axis of the NED frame for the linearized case. In the first two deciseconds, when the quadrotor is commanded to accelerate in the Y direction, the quadrotor has a slight upward acceleration, even though the thrust increment command at time zero is close to zero. This can be explained with the fact that the inner loop control effectiveness inversion is linear, and will add as much RPM on one side of the quadrotor as on the other side to make it bank. Actually, the relation between RPM and thrust is nonlinear (see Figure 4.8), and if all propellers are spinning equally fast, a bank command will therefore result in a slight thrust increase.

More profound is the downward acceleration that happens after half a second, when the quadrotor has to accelerate in the -Y direction. Because the quadrotor is banking to facilitate the acceleration in the +Y direction, the derivative of the vertical acceleration with respect to the bank angle is negative (upward acceleration) for a reduction in bank angle. Therefore, even though eventually around the same thrust is required, initially the thrust is reduced significantly, resulting in a downward acceleration.

Compare this with Figure 4.16, which shows the nonlinear case. Here a larger acceleration is visible in the first two deciseconds. This is caused by the fact that the actuator dynamics are faster than the rotational dynamics. The nonlinear increment is calculated for the tilted thrust vector, therefore a positive thrust increment is commanded by the outer loop INDI controller. However, the rotational dynamics are slower than the thrust dynamics. Therefore, the thrust is increased already before the final attitude is attained. This causes the vehicle to accelerate upwards initially.

After half a second, when the large acceleration change is commanded, the response is quite different from the linear case. Instead of acceleration downward, the vehicle accelerates upward. This can be explained by recognizing that the quadrotor will need more or less the same bank angle to accelerate with the same amount in the other direction. This means that the same thrust is needed. However,

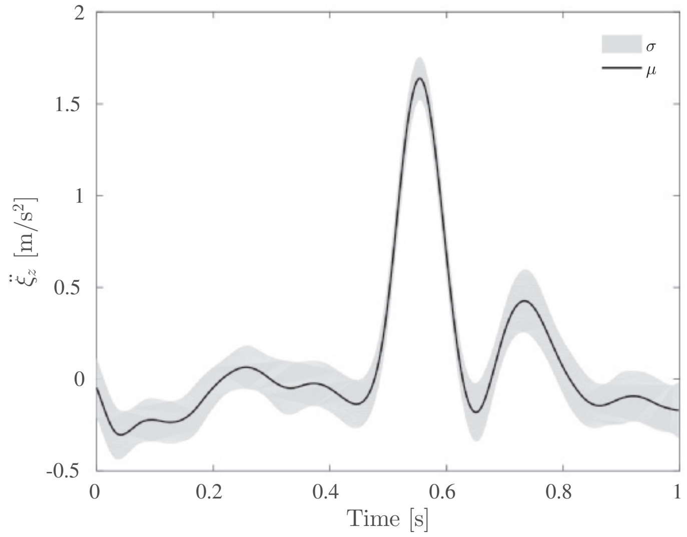


Figure 4.15: Acceleration in the NED Z axis with INDI increments calculated through linearization. From 25 experiments the mean μ is shown, along with one standard deviation σ.

while the vehicle is rotating, it passes the point of zero bank angle, for which it actually needs less thrust to avoid a vertical acceleration. This can explain that the vehicle accelerates upward, reduces thrust, and then overshoots to downward acceleration when it reaches the bank angle at which increased thrust is needed.

Comparing, the nonlinear implementation results in a vertical acceleration that averages better to the intended zero m/s^2. However, there is still some unintended vertical acceleration present. This is mainly attributed to the nonlinear way that input increments are realized, as is described above. Additionally, in the inner loop the nonlinear thrust curve of Figure 4.8 is not taken into account, which may lead to some error.

The acceleration changes in the experiment were the largest possible without introducing saturation in the actuators. Of course, the larger the acceleration change, the larger the nonlinear effects. To analyze if the difference is more significant for larger acceleration changes, more experiments are necessary.

4.8. Conclusions

In this chapter, the control of a micro air vehicle using Incremental Nonlinear Dynamic Inversion (INDI) has been demonstrated for both the inner loop (attitude control) as well as the outer loop (position control) in a cascaded fashion. The disturbance rejection performance of the resultant controller is examined by flying in and out of a 10 m/s wind tunnel flow, showing a more than 7 times lower maximum position deviation than a comparable PID controller. From other experiments it was concluded that the control method is applicable outdoors and that also the control effectiveness of the actuators on the thrust can be adapted online. The nonlinear

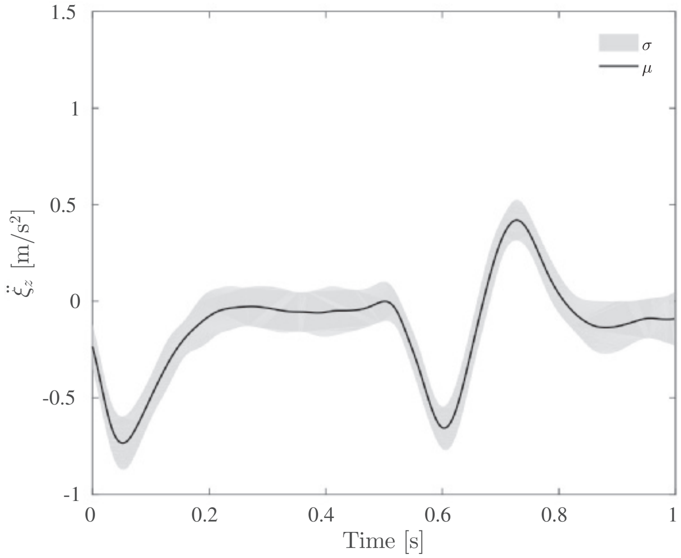


Figure 4.16: Acceleration in the NED Z axis with INDI increments calculated through nonlinear calculation. From 25 experiments the mean μ is shown, along with one standard deviation σ.

calculation of the thrust vector increment reduces the maximum error in vertical acceleration tracking for an aggressive maneuver, but further research is needed to establish if this method yields significant benefits.

The controller derived in this chapter can provide two main benefits. Firstly, the disturbance rejection properties can allow vehicles to operate close to obstacles in gusty environments. Secondly, all parameters, except the position and velocity gains, can be determined based on a test flight and an identification of the actuator dynamics. This makes implementation on new platforms easy and straightforward. Finally, the online adaptation of the control effectiveness can account for changes made to an airframe, and offer even more ease of use.

Future work

The investigation of the effects of linearization in the outer loop indicates that the performance of the inner loop may be further improved by considering the nonlinear relation of rpm and thrust. This could be done by linearly calculating increments for the inner loop, and then use the nonlinear mapping of Figure 4.8 to map the linear increments to the correct nonlinear increments.

Though the inner and outer loop INDI controllers are quite robust, a situation that can still lead to instability is saturation of the actuators. In this case, doing the control allocation through the inverse of the control effectiveness matrix and saturating the resultant control vector, leads to a suboptimal realization of the control objective, because some axes are more important than others. Preliminary research shows that this problem can be solved by taking the axis priorities into account when calculating the control vector [87].

In the derivation of the outer loop INDI controller, it was assumed that changes in ψ would be small, such that the derivative of the thrust vector with respect to ψ could be neglected. A better solution may be to switch to a different Euler angle rotation order. Instead of the common ZYX order, which is used in this chapter, a better choice may be the XYZ order. This will remove any dependency of the thrust vector on the angle ψ.

Furthermore, this control method will be applied to hybrid UAVs, that combine vertical takeoff and landing with fast forward flight using a wing. These vehicles are very prone to be disturbed due to their large aerodynamic surfaces, and INDI is especially good at disturbance rejection. An INDI attitude controller has been used for a tilt rotor vehicle in simulation by Francesco and Mattei [88], but they used a model instead of angular acceleration feedback, which means that the disturbance rejection properties are lost. Preliminary experiments were done with a tailsitter based on the algorithms in this chapter, showing promising results [63].

5

Incremental control and guidance of hybrid aircraft applied to the Cyclone tailsitter UAV

In Chapters 2 to 4 a cascaded INDI controller was developed, analyzed and tested on a quadrotor vehicle. The developed controller performs well against disturbances at the level of attitude control as well as at the level of position control. Furthermore, the INDI attitude controller was extended with a control allocation method that can take priorities into account. It still needs to be shown that this is indeed an effective method for a hybrid MAV.

In this chapter, the developed methodology is extended to a hybrid MAV, taking into account the role of the wing in controlling the acceleration of the vehicle. Making use of scheduling for the control effectiveness matrices, a single INDI controller is developed that covers the entire flight envelope. Through experiments, it is proven that the developed controller is capable of tracking the desired acceleration of the vehicle. Additionally, a solution is provided for active sideslip control and the stability of knife-edge flight is investigated.

This chapter is based on the following article:
Smeur, E. J. J., Bronz, M., de Croon, G. C. H. E., Incremental control and guidance of hybrid aircraft applied to the Cyclone tailsitter UAV. [SUBMITTED] (2018).

5.1. Introduction

Micro Air Vehicles (MAV) are expected to become increasingly more useful, with applications such as mapping, package delivery and meteorological research. Many of these tasks require long endurance, a long range and a high flight speed, which can be achieved by a fixed-wing MAV. On the other hand, operation of these vehicles may involve narrow takeoff and landing sites, such as a ship or urban areas. This requires the versatility of a helicopter, which is able to take off vertically and hover as desired. Hybrid MAVs combine the hovering capability of the helicopter with a wing for fast and efficient forward flight.

Many different hybrid MAVs have been designed, such as quadplanes [30], tilt-wings [21] and tailsitters [12]. These vehicles each have their own advantages and disadvantages, but they have one property that they share: they are difficult to control. More specifically, we see three major challenges for the control of hybrid MAVs: (1) attitude control, (2) velocity control and (3) guidance.

The first challenge is the attitude control of hybrid MAVs. The large flight envelope, often including stalled conditions, makes modeling such a vehicle a difficult and expensive task. Moreover, even if such a model can be found, during flight it may be difficult to obtain the inputs to such a model. For instance, the angle of attack can often not be determined accurately at low airspeed. Furthermore, the large wing surface makes hybrid aircraft particularly susceptible to wind gusts while hovering.

The second challenge is velocity control, with the inputs of attitude and thrust. Here, the main problem is that the forces that can be used to manipulate the acceleration of the vehicle change across the flight envelope. While the thrust is the main controlled force during hover, in forward flight the lift has to be manipulated as well to accommodate accelerations. This is further complicated by wing stall, which again is difficult to model.

The third challenge is the guidance, by which we mean the generation of desired velocities that will lead the MAV to a certain location. A hybrid MAV has a large flight envelope, which means that there are multiple ways of executing certain maneuvers. To stay at one location, it is possible to hover or to make a circle in forward flight. When the vehicle is in forward flight and has to turn around, it is possible to make a turn, or to break, hover, and accelerate in the opposite direction. An important parameter for these decisions is the amount of energy that is expended.

Regarding the challenge of attitude control, some have proposed simple Proportional Integral Derivative (PID) control [13, 89]. Although simple, the accuracy and disturbance rejection capability of this method is limited. Ritz and D'Andrea [90] model the pitch moment as a function of angle of attack and velocity and compensate for this moment in the attitude controller. However, their experimental results show large systematic pitch errors of around 20 degrees. To better deal with changing aerodynamic moments, wind-tunnel measurements can be used [3, 23]. Lustosa et al. performed a wind tunnel campaign to obtain an accurate model, which is used to design a series of LQR controllers that each can control part of the flight envelope. The model relies on the angle of attack and airspeed, two parameters that are difficult to measure onboard the UAV at low speeds. Moreover, wind tunnel measurements are an expensive and time consuming undertaking. Alternatively, the controller can be continuously adapting to the changing vehicle dynamics, even when transitions are performed [24, 91]. The risk of this approach is that, due to disturbances or modeling errors, wrong parameters are learned. The large variance of the learned parameters shown by Knoebel and McLain [91] is likely caused by this phenomenon.

Considering the challenge of velocity control, many papers deal separately with hover, transition, and forward flight [15, 30–32]. Although this approach may produce good results on days without wind, it is not very flexible; for instance when a constant wind requires the vehicle to fly like a fixed wing in order to maintain its position. In such a case, the drone may need to maintain flight in between hover and forward flight. Hartmann *et al.* [21] developed a controller that is able to fly at any airspeed. This allows them to track velocities more accurately and to deal better with wind. However, the controller relies on wind-tunnel data and an extensive trim model.

The last challenge, the guidance, is not discussed in the literature to the best knowledge of the authors. This may be because this topic is not about stability, but about efficiency, and as such it is less essential to achieve flight. Nonetheless, the flight efficiency is still very important, as it is one of the main reasons to choose for a hybrid vehicle instead of a multirotor.

In this chapter, we offer a solution to each of the three challenges using only a minimal amount of modelling. For the attitude and velocity control we propose two cascaded Incremental Nonlinear Dynamic Inversion (INDI) controllers, based on our previous work on INDI for quadrotors [52, 92]. This controller does not need a model of the vehicle's forces and moments, because these can be derived from the acceleration and angular acceleration respectively. Instead, the only required knowledge is the control effectiveness, which is the change in force or moment caused by a change in control input, also known as the control derivatives. The control effectiveness is used in order to calculate increments to the inputs that will result in desired increments in the linear and angular acceleration. The control effectiveness can be obtained through test flights, removing the need for wind-tunnel measurements. For the guidance we propose the heuristic to avoid stall when the current and desired airspeed are above the stall speed. This can save a significant amount of energy, for instance when the vehicle is in forward flight and is required to turn around.

All algorithms developed in this chapter are implemented and tested on the Cyclone tailsitter aircraft shown in Figure 5.1, which was presented in preliminary work [63]. The vehicle was designed for efficiency in forward flight, with up to 90 minutes endurance. The Cyclone is not equipped with a tail or any vertical surface, which causes it to easily pick up a sideslip angle, reducing performance and possibly resulting in loss of lift. In order to avoid this, we include active sideslip control, purely based on accelerometer feedback.

The outline of this chapter is as follows. First, Section 5.2 elaborates on the attitude control using INDI. Then, Section 5.3 deals with estimation and control of

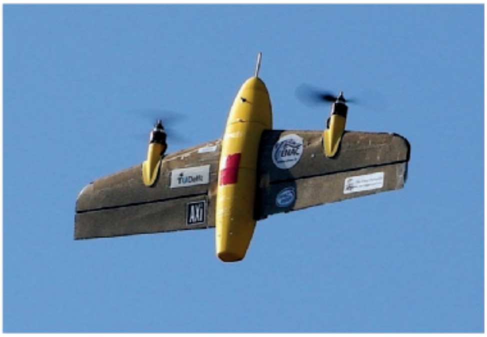


Figure 5.1: The 'Cyclone' hybrid vehicle used in this research.

the sideslip angle. Further, in Section 5.4 the implementation of velocity control using INDI is explained. Section 5.5 discusses the guidance routines developed for the Cyclone. In Section 5.6 results from test flights are discussed. Implementation guidelines are provided in Section 5.7. Finally, in Section 5.8 it is concluded that the designed controller can accurately control a hybrid vehicle within the physical constraints of the actuators, without relying on extensive modeling.

5.2. Attitude Control

Figure 5.2 presents a drawing of the Cyclone along with the body axis definitions. We will refer to yaw, roll and pitch from this perspective, i.e. rotations around the Z, X and Y axes respectively. From Oosedo *et al.* [20], we adopt the ZXY Euler rotation sequence, which is the only sequence used throughout this chapter. The benefit of using the ZXY sequence is that the singularity does not occur at ±90 degrees pitch, but at ±90 degrees roll. Needless to say, the vehicle is intended to visit -90 degrees pitch, whereas this is not the case for ±90 degrees roll.

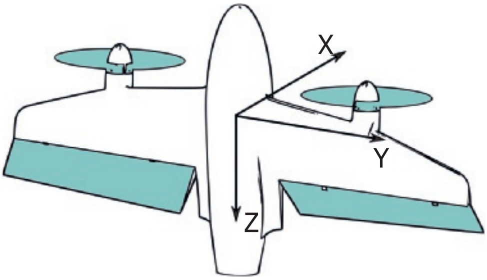


Figure 5.2: The body axis definitions of the *Cyclone*. The four actuators are accentuated with color.

The Cyclone is a hybrid MAV with only four actuators. It has two propellers, which provide the thrust force and the moment around the body X axis. Further, the vehicle has two flaps, which provide moments around the Y and Z axes. The flaps are most effective in forward flight, but even in hover flight the flaps remain effective, because of the airflow coming from the propellers.

In Figure 5.3, the angle of attack and flight path angle are defined. For the performance of the wing, the angle of attack is an important variable.

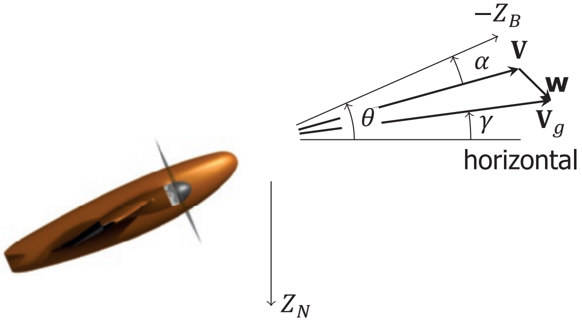


Figure 5.3: Definition of angle of attack α and flight path angle γ. The velocity with respect to the air $\mathbf{V}$ and the velocity with respect to the ground $\mathbf{V}_g$ are connected with the wind vector $\mathbf{w}$.

5.2.1. Center of Gravity

The location of the center of gravity has a large influence on the passive stability of the pitch axis in forward flight. Hover and forward flight require different locations of the center of gravity for passive stability. For stable hover, the center of gravity needs to be aft with respect to the aerodynamic surface. For stable forward flight, the center of gravity needs to be more forward with respect to the aerodynamic surface. Without moving either the center of gravity or the wing during the flight, passive stability in both conditions can not be achieved in a single flight.

The Cyclone is aimed at efficiency in forward flight, which makes carrying around an additional system that changes the center of gravity during flight unattractive. Instead, we opt for a controller that actively controls and stabilizes the attitude across the flight envelope. Still, a compromise needs to be made in terms of the position of the center of gravity with respect to longitudinal stability. Since the aerodynamic moments in forward flight are a lot larger than in hover, because of the higher dynamic pressure, instability in this flight regime is expected to be much harder to control, compared to instability during hover. This is why the center of gravity is placed at the neutral point for forward flight (close to the quarter-chord point). Additionally, the more aft the center of gravity is placed, the closer it is to the flaps, and the smaller the pitch moment that the flaps can generate.

It is clear that a center of gravity that is placed this far to the front, gives rise to a relatively strong pitch-down moment at the high angles of attack (post stall) that can be expected for such a hybrid vehicle. These moments depend not only on the angle of attack itself, but also on the airspeed. Modeling this effect in a windtunnel is time consuming and costly. Furthermore, in real flights both the angle of attack, as well as the airspeed are difficult to measure at low airspeed, which makes any such model difficult to apply. This is why we are looking for a control method that does not rely heavily on knowledge of the airspeed or angle of attack.

5.2.2. Incremental Nonlinear Dynamic Inversion

Incremental Nonlinear Dynamic Inversion (INDI) is an approach driven by the measurement of the angular acceleration [33, 34]. The method is based upon the notion that all moments together, inputs and external moments, produce the angular acceleration that can be measured by deriving it from the gyroscope measurement. If we assume that the external moments do not change rapidly, we can invert for the control effectiveness and calculate an increment in inputs that produces a desired increment in angular acceleration. Then, the angular acceleration is a controlled variable, and it can be used to control angular rates with a simple proportional gain. Similarly, the attitude can then be controlled by setting a certain reference for the angular rates.

A complete derivation and validation of INDI is presented by Smeur *et al.* [52, 92] and is beyond the scope of this chapter. Here, we will briefly summarize the controller.

Consider the input vector to the actuators $\mathbf{u}_A$, consisting of the left and right flaps and the right and left motors, in this order. An increment in inputs causes an increment in angular acceleration and thrust, depending on the control effectiveness matrix $\mathbf{G}$:

$$\begin{bmatrix} \dot{\mathbf{\Omega}} \\ T \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{\Omega}}_0 \\ T_0 \end{bmatrix} + \mathbf{G}(\mathbf{u}_A - \mathbf{u}_{A_0}) \tag{5.1}$$

This can be turned into a control law by simply taking the inverse of $\mathbf{G}$. The controlled variables, the angular acceleration in three axes and the thrust, are now denoted by the virtual control vector $\boldsymbol{\nu}$. The final control law is then:

$$\mathbf{u}_A = \mathbf{u}_{A_0} + \mathbf{G}^{-1}(\boldsymbol{\nu} - \begin{bmatrix} \dot{\mathbf{\Omega}}_0 \\ T_0 \end{bmatrix}) \tag{5.2}$$

where $\dot{\mathbf{\Omega}}_0$ is the measured angular acceleration, T_0 is the current thrust and $\mathbf{G}$ is the control effectiveness matrix. It is possible to add a term that compensates for the effect of propeller inertia on the angular acceleration [52], but this is not taken into account in this chapter. This is because the effect is small compared to the inertia of the vehicle around the Z axis, and the actuator dynamics of the propeller/motor combination are relatively slow.

As the angular acceleration is now a controlled variable, the angular rates can be controlled with simple proportional feedback:

$$\boldsymbol{\nu} = \begin{bmatrix} K_\Omega(\mathbf{\Omega}_{\text{ref}} - \mathbf{\Omega}) \\ T_d \end{bmatrix} \tag{5.3}$$

where T_d is the desired thrust, which is calculated by the outer loop. In practice, as the outer loop is also an INDI controller, it passes the desired thrust increment $T_d - T_0$, which can directly be used in Equation 5.2.

To control the attitude, a second proportional controller is added using feedback of the vector part of the quaternion error:

$$\boldsymbol{\Omega}_{\text{ref}} = K_\eta \begin{bmatrix} q_1^{\text{err}} & q_2^{\text{err}} & q_3^{\text{err}} \end{bmatrix}^T \tag{5.4}$$

where $\mathbf{q}_{\text{err}}$ is given by:

$$\mathbf{q}_{\text{err}} = \mathbf{q}_{\text{ref}} \otimes \mathbf{q}_{\text{s}}^* \tag{5.5}$$

These gains can be tuned, or they can be designed, based on the transfer function of the actuator dynamics [92].

The algorithm is implemented in the Paparazzi open source autopilot software, where commands to actuators are in the range of [-9600, 9600] for servos and [0, 9600] for motors. For the flaps, this maps to a deflection of 30 degrees each way. The servos are attached directly to the flap on the hinge line, so there is no non-linearity from linkages. Because INDI neglects the plant dynamics, but relies heavily on the relation between input and output, it is important to know the position of each actuator at every time. For this purpose, the servo dynamics are modeled as:

$$A(z) = \frac{a}{z - (1 - a)} \tag{5.6}$$

with $a = 0.1$ for a sample frequency of 500 Hz, with a rate limit of 272 degrees per second. For the motors, $a = 0.045$, without a rate limit.

5.2.3. Control Effectiveness Scheduling

Since the flaps are aerodynamic surfaces, their effectiveness depends on the dynamic pressure $q = \frac{1}{2}\rho V^2$, where ρ is the air density and V is the airspeed. The control effectiveness matrix $\mathbf{G}$ therefore changes continuously during flight, mainly depending on the airspeed. Because of the large variation in angle of attack, the best thing would be to have a multi-hole pressure probe to measure the airspeed. The downsides of such a sensor are that it is more expensive, weighs more, and is more difficult to calibrate. Instead, the Cyclone is equipped with a regular Pitot tube, whose direction is fixed for forward flight. Because a Pitot tube needs to be aligned with the airflow to be able to correctly measure the dynamic pressure, the angle of attack needs to be within $\pm$ 30 degrees in order for the Pitot tube to be accurate. This means in practice that it will start to measure the airspeed from 6 m/s or higher. Therefore, an alternative variable must be used for the control effectiveness scheduling whenever the airspeed is low. When the airspeed is too low to be measured, the pitch angle is used as the scheduling variable, leading to a composite control effectiveness function.

The parameters of this function are found by taking segments of the flight data for which the pitch angle and the airspeed (if it can be measured) are relatively constant. For each of these segments, the flap effectiveness is calculated with a linear least squares solution. Afterwards, a quadratic function of airspeed is fitted through the effectiveness values of the segments with measurable airspeed. For parts of the flight when the airspeed can not be measured, a linear function of the pitch angle is used to schedule the control effectiveness. Although the pitch angle only provides limited information, it is an entity that is easy and robust to measure.

The result of this procedure for the Cyclone is the following function:

$$G_{21}(\theta,V)=\begin{cases}(-2.1(1-r_\theta)-4.0r_\theta)\cdot 10^{-3}, & \text{for } V<6\text{m/s}\\ (-2.4-0.031V^2)\cdot 10^{-3}, & \text{for } V\geq 6\text{m/s}\end{cases} \tag{5.7}$$

for the effectiveness on the pitch axis and

$$G_{31}(\theta,V)=\begin{cases}(-2.0(1-r_\theta)-8.0r_\theta)\cdot 10^{-3}, & \text{for } V<6\text{m/s}\\ (-5.6-0.052V^2)\cdot 10^{-3}, & \text{for } V\geq 6\text{m/s}\end{cases} \tag{5.8}$$

for the effectiveness on the yaw axis, where

$$r_\theta=\begin{cases}0, & \text{for } -30\leq\frac{\theta\cdot 180}{\pi}\\ (\frac{\theta\cdot 180}{\pi}+30)/(-30), & \text{for } -60\leq\frac{\theta\cdot 180}{\pi}\leq -30\\ 1, & \text{for } \frac{\theta\cdot 180}{\pi}\leq -60\end{cases} \tag{5.9}$$

and $G_{22}=-G_{21}$, and $G_{32}=G_{31}$. r_θ is defined such that its value is always on the interval [0,1].

For the propeller-motor combination, the control effectiveness around the X axis (roll) did not significantly depend on the airspeed. A good fit of the flight data was obtained just considering the rotational speed of the propeller itself in the control effectiveness. Since we are considering increments, this is analogous to a quadratic relation between propeller rotational speed and produced force. With the chain rule for the derivative, the control effectiveness is found to be:

$$\begin{aligned}G_{13}&=-\mathbf{u}_{A_{f_3}}\cdot 1.8\cdot 10^{-6}\\ G_{14}&=-\mathbf{u}_{A_{f_4}}\cdot 1.8\cdot 10^{-6}\end{aligned} \tag{5.10}$$

5.2.4. Effectiveness of thrust on pitch

During the hover phase, the airflow over the flaps is predominantly generated by the propellers. This means that reducing the total thrust generated by one of the propellers, will have a negative effect on the control effectiveness of the corresponding flap. Especially when descending while hovering (pitch angle close to zero), it can happen that the flow coming from the tail of the airplane dominates the flow of the propellers. In this case the airflow is reversed, and flap deflections will have the opposite effect.

To help cope with this, the minimum thrust level is defined to be 42% when the airspeed is low and there is little flow over the wing ($V<8$ m/s), and 16% otherwise. This will make sure there is always airflow over the flaps.

For this vehicle, the pitch angle is deemed to be the most important degree of freedom to control. However, the aircraft naturally has a pitch down moment. There are cases, with a low airspeed and high angle of attack, when the flaps saturate in their effort to pitch up, without achieving the desired moment. The vehicle can end up 'locked' in this state: trying to pitch up, but in the meantime slowly flying forward.

Because the flaps in this case are already deflected, increasing or decreasing the propeller thrust will affect the speed of the flow over the deflected flaps and as such have a direct effect on the angular acceleration in the pitch axis. This effect is difficult to model exactly, partially because it is a complex function of flap deflection, airspeed and angle of attack. Moreover, increasing the thrust reduces the angle of attack, which also reduces the pitch down moment. Because of the modeling difficulties, this control effectiveness is not taken into account in normal conditions. However, when both flaps are near saturation in an effort to pitch, it is the last resort in order to increase the pitch up moment. In this situation we add a control effectiveness of thrust on pitch of 2.2 rad/s^2 per % thrust for each motor. Since the priority of pitch is higher than that of thrust, as is discussed in Section 5.2.7, the thrust control objective is largely sacrificed in order to pitch up.

5.2.5. Effectiveness of propellers on rotation around Z axis

From test flights, it turned out that the effectiveness of the propellers on the rotation around the body Z axis is limited. A possible explanation is that there is a wing behind the propellers. The wing removes part of the rotation from the propeller slipstream, also known as 'swirl recovery' [93]. In doing so, the wing produces a moment that partly cancels the torque from the propeller. Since the net torque from changing the propeller rotation speed is small compared to the effectiveness of the flaps, we choose to simplify the control effectiveness matrix and neglect this term.

5.2.6. Control effectiveness matrix

Combining the above sections, the final control effectiveness matrix is given by:

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & G_{13}(\mathbf{u}_A) & G_{14}(\mathbf{u}_A) \\ G_{21}(\theta,V) & G_{22}(\theta,V) & G_{23}(\mathbf{u}_A) & G_{24}(\mathbf{u}_A) \\ G_{31}(\theta,V) & G_{32}(\theta,V) & 0 & 0 \\ 0 & 0 & -0.0011 & -0.0011 \end{bmatrix} \tag{5.11}$$

with functions as provided in the sections above.

5.2.7. Control Allocation

Control allocation is an essential part of a hybrid of this design. The reason is firstly that the flaps easily saturate, because of their limited control effectiveness at low airspeed. Secondly, these flaps control the rotation around both the body Y (pitch) and Z (yaw, which would be roll from the airplane perspective) axes, which means that upon saturation, either of these control objectives, or both, will suffer. We make the case here that control around the Y axis is more important, and should therefore have precedence over the control of Z axis.

As has been stated before, the vehicle has a natural tendency to pitch down. This makes returning to hover from forward flight, while maintaining the same altitude, especially tough. In fact, the flaps can remain saturated for multiple seconds while trying to pitch up, making every bit of flap deflection necessary. In this situation, any control effort spent on rotation around the Z axis will reduce the control effort spent on pitching up, making it near impossible to return to hover. Therefore, management of priorities is very important.

Such priorities can be realized with a control allocation method that takes the actuator limits into account.

We have discussed the Weighted Least Squares (WLS) control allocation algorithm [55] in previous work for quadrotor control [87], and apply the same method here. With relative weights for each controlled axis a quadratic programming problem is constructed, which is solved with the active set algorithm. The relative priority factors used for the Cyclone are $[100, 1000, 0.1, 10]$ for rotation around the body X, Y, Z axes, and thrust. The algorithm minimizes a cost function, taking into account the minimum and maximum input increments. The error in the output increment is multiplied by the priority factors, squared and summed to produce the cost function.

Test flights show that the relative priority factors listed above indeed lead to situations where the control of the yaw angle deteriorates when large pitch up moments are needed and saturation occurs. However, even though the yaw angle can be oscillating in these cases, it is not unstable. Therefore, the control allocation makes it possible to use all the control effort to pitch up, enlarging the flight envelope to higher angles of attack.

5.2.8. Knife-edge flight

One difficulty of the tailsitter design, is the landing. As can be seen from Figure 5.2, when the Cyclone is touching the ground it can very easily pitch and fall over. Moreover, a wind gust can provide a large pitching moment on the vehicle when it is standing on the ground, due to the large wing surface and the low center of rotation. Lastly, if there is a constant wind, the Cyclone needs to fly with a considerable pitch angle to keep its position, while in the end it needs to stand upright on the ground. Combined, these things make taking off, but especially landing a challenging endeavour.

A partial solution to this could be to align the airspeed vector with the body ZY plane (rolling to gain airspeed instead of pitching, or flying 'knife-edge'). The benefit of doing this, is that the lateral surface of the Cyclone is much smaller than the frontal surface. This means that the roll angle needed to maintain a certain airspeed, is considerably smaller than the pitch angle needed for the same airspeed. Finally, with a smaller angle, it is easier to land.

Additionally, knife-edging is not expected to lead to large constant moments that need to be countered by the flaps, like is the case when flying at large angle of attack. This way, the controllability could be improved when flying at low speeds.

To evaluate if this is truly a useful concept, we need to consider the stability in this flight mode. Assume that the Cyclone, in the lateral axis, can be modeled as a flat plate. Also assume that the roll angle, because of the low sideways drag, is small when knife-edging.

The center of gravity of the Cyclone, seen from the side, is in the middle of

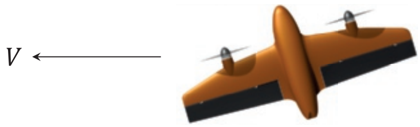


Figure 5.4: Knife-edge flight: hovering with a roll angle to attain a sideways velocity.

the aircraft. For a flat plate, the center of pressure is located at the quarter chord point [94]. Therefore, for a flat plate to be stable, the center of gravity needs to be in front of the quarter chord point. Since this is not the case for the Cyclone, the knife-edge maneuver, based on this (crude) analysis, is not passively stable. However, it is still possible that the control system is able to cope with this.

To verify this in practice, an experiment is performed. Given that the controlled vehicle is stable in an indoor environment in all axes, the hypothesis is that while knife-edging, the vehicle is not stable around the body Z axis. On a windy day, the Cyclone is commanded to keep his position, while not changing the heading. The operator sets the heading to be orthogonal to the wind direction, such that the Cyclone is knife-edging. Since altitude changes influence the stability, the experiment is performed at constant altitude. From a flight prior to the experiment, executed at constant airspeed, it was concluded from the ground speed that the wind was around 4 to 5 m/s.

Figure 5.5 shows the ψ angle for the experiment. Large errors in the ψ angle occur repeatedly while the vehicle is hovering for 45 seconds. Each of the peaks in the figure is preceded by saturation of one of the flaps, which indicates that the maximum yaw control effort is reached. From these results, it is concluded that the control system is not able to stabilize the vehicle in a knife-edge maneuver. As such, this maneuver is not integrated in the autonomous flight algorithm of the Cyclone.

5.3. Sideslip

In the design of the Cyclone, efficiency and simplicity are major design drivers. Since the vehicle already has two propellers to provide moments around the body X axis, there is no need for a vertical stabilizer or a rudder. The vehicle is capable of fully controlling its attitude, using the four actuators that it has. The benefit of not having a vertical stabilizer is twofold. First, it reduces the susceptibility to wind gusts while hovering, as there is less aerodynamic surface. Second, a tail would add to the structural weight, and it would produce additional drag.

However, even though the propellers may be able to stabilize and control the rotation around the body X axis in forward flight, there is the important restriction that for the wing to provide lift efficiently, the sideslip angle should be small, ideally zero. Actively controlling the sideslip to zero requires a measurement or an estimate of the sideslip angle.

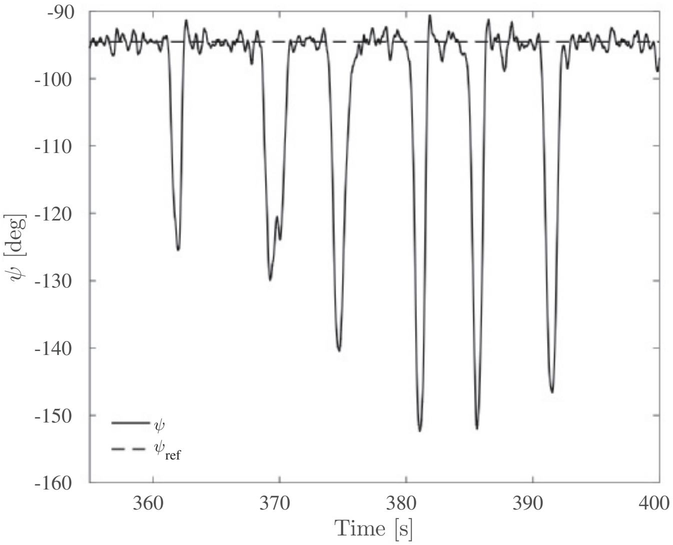


Figure 5.5: The ψ angle for the knife-edge experiment.

5.3.1. Estimating the sideslip angle

Aerodynamic forces are typically defined in the wind frame, which has its origin in the center of gravity of the aircraft. The X axis points in the direction of the airspeed vector, the Z axis lies in the plane of symmetry of the aircraft, positive below the aircraft, and the Y axis follows from the right hand rule. The angle of attack and the sideslip angle are the rotations from the body frame to the wind frame.

When the vehicle is slipping, the airspeed vector has a component in the body Y axis. Since the drag D is in the same direction as the airspeed vector, the drag has a component in the Y axis D_y:

$$\sin\beta = \frac{-D_y}{D} \tag{5.12}$$

The component D_y is measured by the accelerometer as specific force $f_y = D_y/m$. The total amount of drag is given by:

$$D = C_D\frac{1}{2}\rho V^2 S \tag{5.13}$$

where C_D is the drag coefficient of the vehicle.

Combining Equations 5.13 and 5.12, using a small angle approximation for the sine, combining all constant parameters into c_1 and adding a bias compensation b_1 we can write:

$$\beta = c_1\frac{f_y}{V^2} + b_1 \tag{5.14}$$

In order to find an estimate for the parameters b_1 and c_1, a sideslip vane was mounted on the airframe, as can be seen in Figure 5.7. A flight was made without

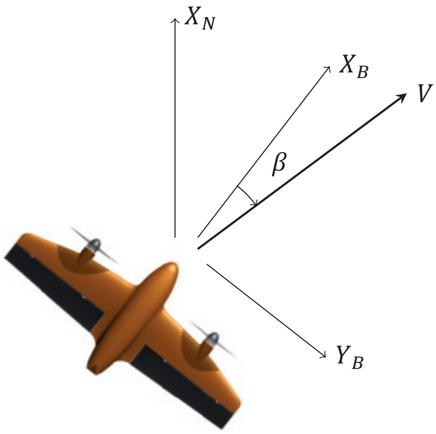


Figure 5.6: Definition of the sideslip angle β, seen from above.

proper sideslip control, such that there was plenty of sideslip to estimate. The specific force f_y is filtered with a second order Butterworth filter with a cutoff frequency of 5 Hz. The day of this test flight the wind was negligible, which is why for this flight, it is acceptable to use the norm of the GPS speed as a measurement for the airspeed V. From the test flight, a section of 200 seconds was chosen which only contains forward flight, with the GPS flight speed as shown in Figure 5.9. The data was divided in a training set (first 80%) and a test set (second 20%). A linear least squares fit of Equation 5.14 on the training set gives a root mean square (RMS) error on the test set of 0.1189 rad.

This equation contains a division with V^2, which means that at low airspeeds, this equation will become quite unstable. This is because the sideslip angle is ill-defined when the airspeed is zero. However, for the purpose of β feedback, a signal is preferred that remains stable when the airspeed approaches zero. Therefore, a further simplification is suggested, removing the dependency on the airspeed:

$$\beta = c_2 f_y + b_2 \tag{5.15}$$

which gives an even lower RMS error of 0.1122 rad for the same test set. Both fits are shown in Figure 5.8, along with the measurement from the sideslip vane. For the feedback control, the simpler and more robust Equation 5.15 is selected. Note that the best fit for the dataset is obtained by dividing by V instead of V^2 with an RMS error of 0.0757, though this is still not robust when the airspeed approaches zero.

5.3.2. Sideslip control

Now that an estimate of the sideslip angle is available, without any need for a sideslip vane, this estimate can be used to change the reference heading such that

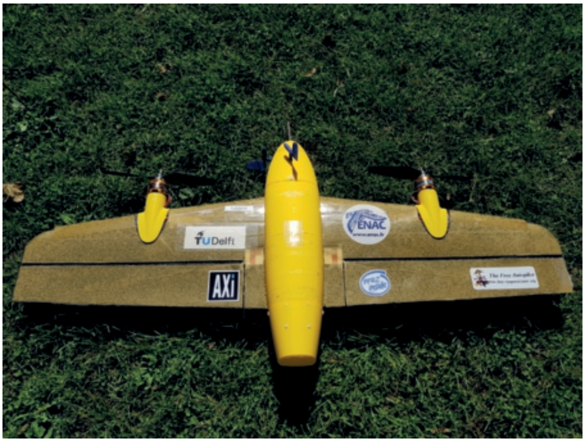


Figure 5.7: The Cyclone with angle of attack and sideslip vanes mounted. The vanes are only used for validation and not for control.

the sideslip is removed. The way this is done, is by setting the rate of change of the reference heading angle proportional to the sideslip angle with a gain K_β. Added to this is the feed forward component to make a coordinated turn [95]:

$$\dot{\psi}_{\text{ref}} = \frac{g\tan(\phi_t)}{V_l} + K_\beta \beta \tag{5.16}$$

where $\dot{\psi}_{\text{ref}}$ is the rate of change of the heading reference, g is the gravitational constant and V_l is a limited airspeed, with 10 m/s as a lower limit, to avoid unachievable rotations.

ϕ_t is defined equal to ϕ_{ref}, except when $\theta_{\text{ref}} > 0$ and $|\phi_{\text{ref}}| < \theta_{\text{ref}}$, then $\phi_t = \text{sign}(\phi_{\text{ref}})\theta_{\text{ref}}$. The reason for this is that the airfoil is not designed for inverted flight, so when pitching backward, the vehicle should yaw around and align itself with the direction of motion. The pitch angle reference is limited such that the maximum is 25° (pitching backward), as the vehicle appears to not be very stable at high positive pitch angles. Combined, the result is that commanding the Cyclone to fly to a waypoint towards the rear of the drone, from a hovering position, first leads to it pitching back a maximum of 25 degrees, while yawing around. Gradually, the vehicle orients itself with the direction of motion, allowing it to transition into forward flight.

5.4. Velocity Control

The velocity of the Cyclone can be controlled, by controlling the linear acceleration of the vehicle. This can be done by applying the INDI methodology, as we have shown in a previous paper [53]. In that paper, we showed that increments in linear acceleration can be achieved by increments in the thrust vector of a multirotor helicopter. For the Cyclone, the thrust vector is used to control the linear accelerations as well, but additionally it uses the lift force generated by the wing.

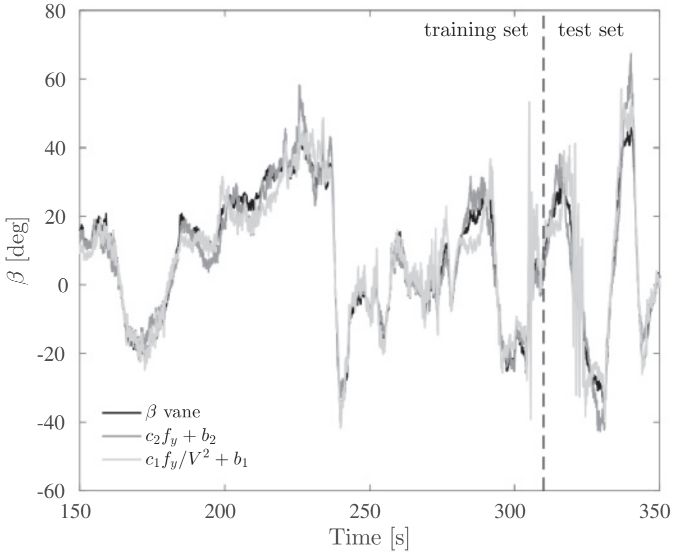


Figure 5.8: The sideslip angle during an identification flight, along with a fit of f_y and f_y/V^2.

Therefore, the controller developed previously has to be amended, such that the control derivatives of the wing are taken into account.

One of the reasons to use INDI is that no precise model is required, as model inaccuracies are compensated by the fast increments. In the following derivation of the control derivatives, quite a few simplifying assumptions are made for two reasons. First, these assumptions lead to a controller that does not rely on aerodynamic angles that are hard to measure or estimate accurately at low airspeed, such as the angle of attack. Second, the assumptions keep the model simple, such that it is easy to implement on different vehicles.

With this in mind, consider the equation that describes the acceleration of a hybrid MAV in the North East Down (NED) frame:

$$\ddot{\boldsymbol{\xi}} = \mathbf{g} + \frac{1}{m}\mathbf{L}_N(\boldsymbol{\eta}, V) + \frac{1}{m}\mathbf{D}_N(\boldsymbol{\eta}, V) + \frac{1}{m}\mathbf{T}_N(\boldsymbol{\eta}, T) \tag{5.17}$$

where $\ddot{\boldsymbol{\xi}}$ is the second derivative of the position, $\mathbf{g}$ is the gravity vector, and m the mass of the vehicle. Further, $\mathbf{L}_N(\boldsymbol{\eta}, V)$ is the lift vector, $\mathbf{D}_N(\boldsymbol{\eta}, V)$ is the drag vector, and $\mathbf{T}_N(\boldsymbol{\eta}, T)$ is the thrust vector.

First, in order to have control derivatives, a representation of the attitude $\boldsymbol{\eta}$ needs to be established. Here, the choice is made to work with Euler angles, because it is a concise set that is easy to work with. Again, we work with the ZXY rotation order, such that there is no singularity at -90 degrees pitch. For this representation, the rotation matrix from the body axes to NED axes is:

$$\mathbf{M}_{NB} = \begin{bmatrix} \mathrm{c}\theta\mathrm{c}\psi - \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi & -\mathrm{c}\phi\mathrm{s}\psi & \mathrm{s}\theta\mathrm{c}\psi + \mathrm{s}\phi\mathrm{c}\theta\mathrm{s}\psi \\ \mathrm{c}\theta\mathrm{s}\psi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{c}\psi & \mathrm{c}\phi\mathrm{c}\psi & \mathrm{s}\theta\mathrm{s}\psi - \mathrm{s}\phi\mathrm{c}\theta\mathrm{c}\psi \\ -\mathrm{c}\phi\mathrm{s}\theta & \mathrm{s}\phi & \mathrm{c}\phi\mathrm{c}\theta \end{bmatrix} \tag{5.18}$$

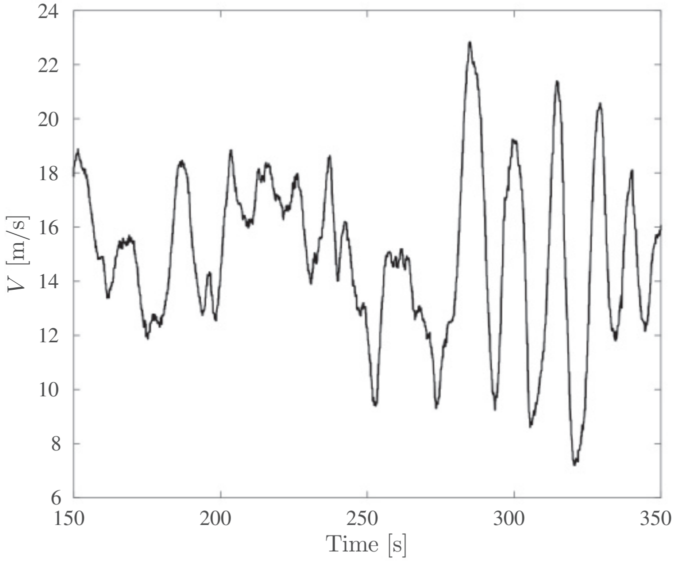


Figure 5.9: Ground speed during the manually piloted sideslip identification flight. There was no perceptible wind during the flight.

Where the sine and the cosine are abbreviated with the letters s and c respectively. The thrust vector in the NED frame is now simply:

$$\mathbf{T}_N = \mathbf{M}_{NB}\mathbf{T}_B = \begin{bmatrix} (\mathrm{s}\theta\mathrm{c}\psi + \mathrm{s}\phi\mathrm{c}\theta\mathrm{s}\psi)T \\ (\mathrm{s}\theta\mathrm{s}\psi - \mathrm{s}\phi\mathrm{c}\theta\mathrm{c}\psi)T \\ \mathrm{c}\phi\mathrm{c}\theta T \end{bmatrix} \tag{5.19}$$

The lift vector is typically defined orthogonal to the airspeed vector, in the body XZ plane. In order not to introduce a dependency on aerodynamic angles, we assume the flight path angle and the sideslip to be small. In general, missions are expected not to have large flight path angles, so this should be a valid assumption. This means that the direction of the lift vector does not depend on the pitch angle, and the vector should not be rotated with the pitch angle ($\mathbf{M}_{NB}^{\theta=0}$). The amount of lift does depend on the pitch angle, since the angle of attack is equal to the pitch angle if the flight path angle is small:

$$\mathbf{L}_N = \mathbf{M}_{NB}^{\theta=0}\mathbf{L}_B^{\theta=0}(\theta, V) = \begin{bmatrix} \mathrm{s}\phi\mathrm{s}\psi L(\theta, V) \\ -\mathrm{s}\phi\mathrm{c}\psi L(\theta, V) \\ \mathrm{c}\phi L(\theta, V) \end{bmatrix} \tag{5.20}$$

where $L(\theta, V)$ describes the magnitude of the lift vector as a function of pitch and airspeed. In short, the magnitude of the lift depends on the pitch angle, but the direction of the lift vector is indifferent to the pitch angle.

Since we assume zero flight path angle, for the drag it also holds that the direction is indifferent to the pitch angle. The drag force is then simply given by:

$$\mathbf{D}_N = \mathbf{M}_{NB}^{\theta=0}\mathbf{D}_B^{\theta=0}(\theta, V) = \begin{bmatrix} \mathrm{c}\psi D(\theta, V) \\ \mathrm{s}\psi D(\theta, V) \\ 0 \end{bmatrix} \tag{5.21}$$

The next step is to take partial derivatives of these forces, in order to obtain control derivatives. In order predict how the acceleration is going to change, a first order Taylor expansion of Equation 5.17 is applied:

$$\begin{aligned} \ddot{\boldsymbol{\xi}} = \; & \mathbf{g} + \tfrac{1}{m}\mathbf{L}_N(\boldsymbol{\eta}_0, V_0) + \tfrac{1}{m}\mathbf{D}_N(\boldsymbol{\eta}_0, V_0) + \tfrac{1}{m}\mathbf{T}_N(\boldsymbol{\eta}_0, T_0) \\ & + \tfrac{\partial}{\partial\phi}\tfrac{1}{m}\mathbf{L}_N(\phi, \theta_0, \psi_0, V_0)|_{\phi=\phi_0}(\phi - \phi_0) \\ & + \tfrac{\partial}{\partial\theta}\tfrac{1}{m}\mathbf{L}_N(\phi_0, \theta, \psi_0, V_0)|_{\theta=\theta_0}(\theta - \theta_0) \\ & + \tfrac{\partial}{\partial\psi}\tfrac{1}{m}\mathbf{L}_N(\phi_0, \theta_0, \psi, V_0)|_{\psi=\psi_0}(\psi - \psi_0) \\ & + \tfrac{\partial}{\partial V}\tfrac{1}{m}\mathbf{L}_N(\phi_0, \theta_0, \psi_0, V)|_{V=V_0}(V - V_0) \\ & + \tfrac{\partial}{\partial\theta}\tfrac{1}{m}\mathbf{D}_N(\theta, \psi_0, V_0)|_{\theta=\theta_0}(\theta - \theta_0) \\ & + \tfrac{\partial}{\partial\psi}\tfrac{1}{m}\mathbf{D}_N(\theta_0, \psi, V_0)|_{\psi=\psi_0}(\psi - \psi_0) \\ & + \tfrac{\partial}{\partial V}\tfrac{1}{m}\mathbf{D}_N(\theta_0, \psi_0, V)|_{V=V_0}(V - V_0) \\ & + \tfrac{\partial}{\partial\phi}\tfrac{1}{m}\mathbf{T}_N(\phi, \theta_0, \psi_0, T_0)|_{\phi=\phi_0}(\phi - \phi_0) \\ & + \tfrac{\partial}{\partial\theta}\tfrac{1}{m}\mathbf{T}_N(\phi_0, \theta, \psi_0, T_0)|_{\theta=\theta_0}(\theta - \theta_0) \\ & + \tfrac{\partial}{\partial\psi}\tfrac{1}{m}\mathbf{T}_N(\phi_0, \theta_0, \psi, T_0)|_{\psi=\psi_0}(\psi - \psi_0) \\ & + \tfrac{\partial}{\partial T}\tfrac{1}{m}\mathbf{T}_N(\phi_0, \theta_0, \psi_0, T)|_{T=T_0}(T - T_0) \end{aligned} \tag{5.22}$$

The first terms can be simplified to the current acceleration:

$$\mathbf{g} + \frac{1}{m}\mathbf{L}_N(\boldsymbol{\eta}_0, V_0) + \frac{1}{m}\mathbf{D}_N(\boldsymbol{\eta}_0, V_0) + \frac{1}{m}\mathbf{T}_N(\boldsymbol{\eta}_0, T_0) = \ddot{\boldsymbol{\xi}}_0 \tag{5.23}$$

This term captures all of the forces acting on the drone and can be obtained by adding the gravity vector to the acceleration measurement (in the NED frame). The other terms describe changes to this sum of forces due to changes in attitude, velocity and thrust.

The variable ψ is not free to choose, as it is used for the control of the sideslip. We assume that changes in ψ are small, such that we can neglect those terms. From analysis of test flight data, we concluded that changes in the drag are generally small compared to the other terms. This leaves us with the following equation:

$$\ddot{\boldsymbol{\xi}} = \ddot{\boldsymbol{\xi}}_0 + \frac{1}{m}\left(\mathbf{G}_T(\boldsymbol{\eta}_0, T_0) + \mathbf{G}_L(\boldsymbol{\eta}_0, V)\right)\left(\mathbf{u}_c - \mathbf{u}_{c_0}\right) \tag{5.24}$$

where $\mathbf{u}_c = [\phi \;\; \theta \;\; T]^T$, and the control effectiveness matrices are given by:

$$\mathbf{G}_T(\boldsymbol{\eta}, T) = \begin{bmatrix} \mathrm{c}\phi\mathrm{c}\theta\mathrm{s}\psi T & (\mathrm{c}\theta\mathrm{c}\psi - \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi)T & \mathrm{s}\theta\mathrm{c}\psi + \mathrm{s}\phi\mathrm{c}\theta\mathrm{s}\psi \\ -\mathrm{c}\phi\mathrm{c}\theta\mathrm{c}\psi T & (\mathrm{c}\theta\mathrm{s}\psi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{c}\psi)T & \mathrm{s}\theta\mathrm{s}\psi - \mathrm{s}\phi\mathrm{c}\theta\mathrm{c}\psi \\ -\mathrm{s}\phi\mathrm{c}\theta T & -\mathrm{c}\phi\mathrm{s}\theta T & \mathrm{c}\phi\mathrm{c}\theta \end{bmatrix} \tag{5.25}$$

and

$$\mathbf{G}_L(\boldsymbol{\eta}, V) = \begin{bmatrix} c\phi s\psi L(\theta,V) & s\phi s\psi \frac{\partial}{\partial\theta} L(\theta,V) & 0 \\ -c\phi c\psi L(\theta,V) & -s\phi c\psi \frac{\partial}{\partial\theta} L(\theta,V) & 0 \\ -s\phi L(\theta,V) & c\phi \frac{\partial}{\partial\theta} L(\theta,V) & 0 \end{bmatrix} \tag{5.26}$$

Now, what is left is to define the functions $L(\theta, V)$ and $\frac{\partial}{\partial\theta} L(\theta, V)$. Unfortunately, we do not have a proper aerodynamic model. Nonetheless, it can be recognized that, again assuming zero flight path angle, gravity will have to be compensated by a combination of thrust and lift from the wing. Therefore, we simply employ the following function:

$$L(\theta, V) \approx L(\theta) = -9.81 \sin(-\theta) m \tag{5.27}$$

where θ is bounded between $-\pi/2$ and 0. This is the function that is used in the test flights presented in this chapter, but it would be more accurate to divide this function by the cosine of ϕ, to reflect the additional lift that needs to be produced in a turn.

Similarly, we assume that in forward flight the thrust just compensates the drag, and its effect on accelerations other than in the thrust axis is small, such that for T_0 in Equation 5.24 we can write:

$$T(\theta) = -9.81 \cos(\theta) m \tag{5.28}$$

where again θ is bounded between $-\pi/2$ and 0.

For the derivative of the lift with respect to the pitch angle, it is not possible to use equation 5.27, because even though through the flight control system the lift will be close to this equation, it is too much simplified to obtain the control derivatives from it. Using test flights, we fit the derivate of the angle of attack, measured with an α vane, with the derivative of the measured acceleration at several flight conditions. The best fit is obtained with an α vane, though it may be possible to estimate the control effectiveness using the pitch angle instead of angle of attack, if such a vane is not available. The effectiveness at these flight conditions is subsequently approximated with the following function:

$$\frac{\partial}{\partial\theta} L(\theta, V) = \begin{cases} -24.0 r_\theta m, & \text{for } V < 12\text{m/s} \\ -(V - 8.5) \cdot 6.88 m, & \text{for } V \geq 12\text{m/s} \end{cases} \tag{5.29}$$

where

$$r_\theta = \begin{cases} 0, & \text{for } -40 \leq \frac{\theta \cdot 180}{\pi} \\ (\frac{\theta \cdot 180}{\pi} + 40)/(-40), & \text{for } -80 \leq \frac{\theta \cdot 180}{\pi} \leq -40 \\ 1, & \text{for } \frac{\theta \cdot 180}{\pi} \leq -80 \end{cases} \tag{5.30}$$

5.4.1. Effectiveness of the flaps on the lift

The flaps, whose purpose it is to control the rotations around the body Y and Z axes, also have a significant effect on the produced lift. This situation is depicted schematically in Figure 5.10, and holds for hover as well as forward flight. In order to achieve a desired acceleration, the vehicle needs to increase or decrease the pitch angle. The required flap deflections for this change in pitch angle, initially lead to an acceleration in the opposite direction, because it increases the lift the flap produces. In linear time invariant systems theory, this is commonly referred to as undershoot, which is caused by non-minimum phase zeros [96]. This opposite reaction gives rise to oscillations in the desired pitch angle.

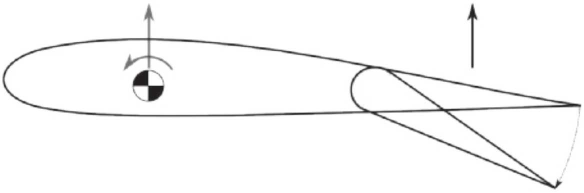


Figure 5.10: Deflecting a flap downward immediately increases the lift the flap produces, as is indicated with the black arrow. The resulting force and moment in the center of gravity are drawn in grey. Ultimately, when the vehicle pitches down the lift decreases.

Whenever the controller commands a pitch change in order to change the acceleration, the vehicle at first accelerates in the direction opposite to the desired one. This will cause the controller to increase the command, even though the original command would have lead to the correct acceleration over time. This leads to oscillations, both in forward flight and while hovering, which is observed in test flights. Such an oscillation of the acceleration in body X axis is shown for hover in Figure 5.11. Along with the measurement, a simple linear least squares fit is depicted using as inputs an offset, the pitch rate and the pitch, according to the model below:

$$acc = [\,1 \quad q \quad \theta\,]\mathbf{B}_1 \tag{5.31}$$

where q is the pitch rate and $\mathbf{B}_1$ is a vector of coefficients. These inputs clearly can not explain the measured data, as the fitted data does not coincide with the measured data at all. That is why the flap deflection has to be added to this model, as is shown below:

$$acc = [\,1 \quad q \quad \theta \quad u_{f_0} \quad u_{f_1}\,]\mathbf{B}_2 \tag{5.32}$$

with $\mathbf{B}_2$ a different vector of coefficients. With this model, the fit is much better, as can be observed from Figure 5.11. From this we can conclude that the flap deflection indeed plays a large role in the lift production.

To cope with this effect, a possible solution is to increase the control effectiveness of pitch on the acceleration in the controller. Since the controller gain is the inverse of the control effectiveness, this reduces the initial control effort, such that further increments in pitch angle are needed to achieve the correct acceleration. This method was tried out in practice, and by scaling the control effectiveness of the pitch on the acceleration by a factor two it was possible to remove the oscillation during hover flight.

Although these results are encouraging, the influence of modifying the control effectiveness from its true value on the flight performance is not yet well understood. Therefore, in this section we present a second solution. The concept is that

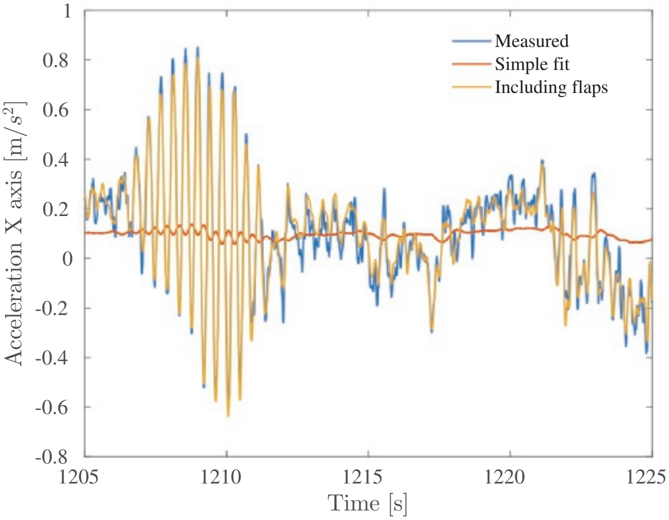


Figure 5.11: Acceleration in the body X axis, along with model fits using input data.

if the vehicle does not react to accelerations caused by movement of the flaps, the problem of oscillations is removed. To achieve this, the acceleration caused by the flaps is modeled, high pass filtered, and subtracted from the acceleration measurement. The reason to apply the high pass filter is that continuous flap deflections may cause offsets in the acceleration measurement, and only the transient flap movements need to be accounted for.

The compensated acceleration $\ddot{\xi}_{\text{comp}}$ is calculated as follows:

$$\ddot{\xi}_{\text{comp}} = \ddot{\xi}_f - \mathbf{M}_{NB} \begin{bmatrix} H_h(-u_{f_0} + u_{f_1})G_{\text{flap}} \\ 0 \\ 0 \end{bmatrix} \tag{5.33}$$

where H_h is the high pass filter, u_{f_0} is the left flap deflection and u_{f_1} the right flap deflection, both low pass filtered to synchronize them with the low pass filtered acceleration measurement. The effectiveness of the sum of flap deflections on the acceleration in the body X axis is denoted by G_{flap}. The high pass filter used is a fourth order Butterworth filter with the cutoff frequency tuned to be 0.5 Hz. The outer loop INDI control scheme, including the flap effectiveness compensation, is depicted in Figure 5.12.

Both methods presented in this section succeed in removing the oscillation in test flights. Arguably, it is a lot simpler to modify the control effectiveness, than to implement the compensation for the flap effectiveness. On the other hand, because of modifying the control effectiveness, it may take longer to counteract disturbances. This is not expected to be a problem for the compensation approach. A detailed analysis of the performance of both methods is a topic for future research.

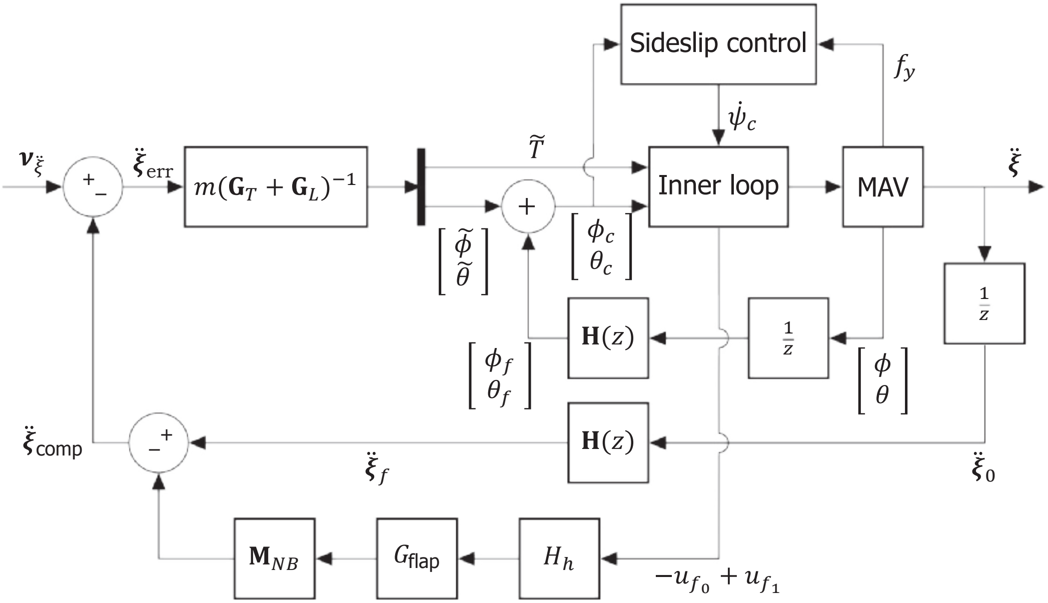

Sideslip control
f_y
$\dot{\psi}_c$
$\nu_{\ddot{\xi}}$
$\ddot{\xi}_{\text{err}}$
$m(\mathbf{G}_T + \mathbf{G}_L)^{-1}$
$\widetilde{T}$
Inner loop
MAV
$\ddot{\xi}$
$\begin{bmatrix}\widetilde{\phi}\\ \widetilde{\theta}\end{bmatrix}$
$\begin{bmatrix}\phi_c\\ \theta_c\end{bmatrix}$
$\mathbf{H}(z)$
$\frac{1}{z}$
$\begin{bmatrix}\phi_f\\ \theta_f\end{bmatrix}$
$\begin{bmatrix}\phi\\ \theta\end{bmatrix}$
$\ddot{\xi}_{\text{comp}}$
$\ddot{\xi}_f$
$\ddot{\xi}_0$
$\mathbf{M}_{NB}$
G_{flap}
H_h
$-u_{f_0} + u_{f_1}$

Figure 5.12: The outer INDI control structure, including the flap effectiveness compensation. Euler angles have the ZXY rotation order.

5.4.2. Attitude gains in forward flight

The roll (body X axis) and pitch (body Y axis) are controlled with different actuators with different dynamics, so they allow for different control gains. The propellers react slower than the flaps, which means that the K_η gain should be lower for the roll axis than for the pitch axis. The result is that the vehicle is more aggressive in the pitch axis than in the roll axis.

For the hover scenario this is generally not a problem, but for turns in forward flight this can lead to altitude errors. In a turn, a continuous combination of roll (around the X axis in Figure 5.2) and pitch rate is necessary. This is realized as the attitude reference rotates ahead of the actual attitude. The error in roll and pitch angle, multiplied with the K_η gain, produces the reference roll and pitch rate. With a higher gain on the pitch axis, the vehicle pitches up disproportionally. As the vehicle is pitching up more, the attitude error becomes smaller in the pitch axis, and the roll and pitch rates are in proportion again. The increased pitch angle leads to a higher lift than intended, and the vehicle ascends.

This is observed in real flights, especially when making long, almost 180° turns with a pitch gain of 13.3 and a roll gain of 7.6, as is shown for two flights in the left plot of Figure 5.13. Most notable is the increase in altitude of more than 18 m during the turn, but also a pitch angle error exists during the turn, while there is no error in the tracking of angular rates. To cope with this effect, both K_η gains are given the same value of 7.6 whenever the Cyclone flies faster than 12 m/s. Two more turns were made with these equal gains, the results of which are shown in the right plot of Figure 5.13. Now, the altitude error stays within 2 meters of the desired 40 m altitude.

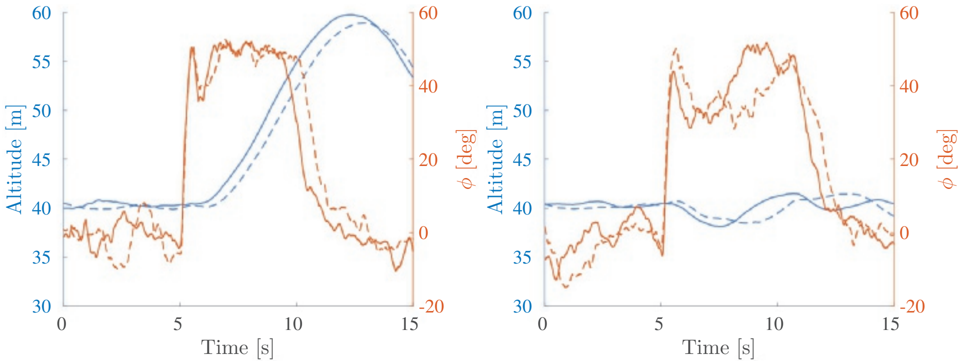


Figure 5.13: Four turns of almost 180° heading change: two with a high pitch gain (left), and two with equal pitch and roll gains (right). Target altitude is 40 m.

5.5. Guidance

In principle, the Cyclone can be guided like any INDI controlled multirotor. This is described in previous research [53], and starts from a certain position error. Multiplied with a gain this produces the desired velocity vector. Subtract the actual ground velocity, multiplied with a gain, and the desired acceleration is obtained. Basically, this is a PD controller that provides the acceleration reference for the INDI controller as is shown below:

$$\boldsymbol{\nu}_{\ddot{\xi}} = ((\boldsymbol{\xi}_{\text{ref}} - \boldsymbol{\xi})K_{\xi} - \dot{\boldsymbol{\xi}})K_{\dot{\xi}} \tag{5.34}$$

5.5.1. Efficient turning

During initial test flights, this was the method of guidance for the Cyclone. Though this method is feasible, there is a specific downside to this approach. This form of guidance will always result in the shortest ground track, which is not necessarily the most efficient for a hybrid vehicle. Take the example of the vehicle cruising at 20 m/s with a certain heading. If the vehicle is now commanded to fly in the opposite direction, it will have to break all the way to 0 m/s, and then accelerate back to 20 m/s in the opposite direction. Since the Cyclone is less efficient while hovering, this approach is expected to be less efficient than a turn at the same airspeed.

To cope with this, a rule based strategy is employed, depending on the current airspeed and the desired airspeed. If the current airspeed is higher than 10 m/s and the desired airspeed is higher than 14 m/s, the vehicle will make a turn (fixed wing style). In this case, the airspeed is controlled, also during the turn, and if applicable the drone will accelerate or decelerate during the turn. The reason that the desired airspeed has to be larger than 14 m/s, is that above this airspeed, the measurement of the airspeed is considered very reliable, avoiding any kind of switching behaviour. In all other cases, the vehicle will take the direct approach, as explained above.

5.5.2. Approaching a waypoint

The desired velocity towards a goal position is calculated by a proportional controller. This works well if the velocity towards the waypoint is low, but if it is large, it may lead to overshoot. The reason is that the proportional gain will require a certain deceleration per meter, which means that if the speed towards the waypoint is high, the required deceleration is high. However, the maximum deceleration of the Cyclone is limited, hence the overshoot.

If we assume this maximum deceleration to be constant over the flight envelope, we can calculate the maximum allowable speed as a function of the distance at which the deceleration is started. Using classical mechanics, we arrive at $v = \sqrt{2da_{\text{max}}}$, where d is the distance to the waypoint and a_{max} is the maximum deceleration. If the speed commanded by the proportional controller is higher than this maximum allowable speed, it is reduced to the maximum allowable speed. This avoids overshoot and limits pitch-up problems during the transition to hover. Specifically, it enables the vehicle to approach a waypoint with tailwind, without any overshoot.

5.5.3. Line following

In order to give some kind of guarantee on the path that the vehicle will follow, some kind of path tracking algorithm is necessary. Eventually the goal is to make the Cyclone able to track any path that is within the performance limits of the vehicle, possibly using a method from literature [97]. For now, the Cyclone has straightforward line following functionality, as most paths can be approximated with a combination of lines.

Each line is defined with a start and end point, and a corresponding field of ground velocity vectors is calculated that converges to and along the line segment. The angle λ of these vectors with respect to the line is given by:

$$\lambda = \text{atan}\left(\frac{d + 0.05d^2}{50}\right) \tag{5.35}$$

where d is the absolute distance orthogonal to the line. The resultant vector field for a fixed speed is shown in Figure 5.14.

The magnitude of the velocity vectors can be predefined, or they can be proportional to the distance to the end point. By setting their magnitude to a relatively large number, while limiting the maximum airspeed, the Cyclone can be made to fly constantly at this airspeed. The line following is only valid if the normal line that goes through the end point is not crossed, and the vehicle is a minimum distance away from the end point. At this point, the vehicle switches to the next element in the flight plan, which could be another line.

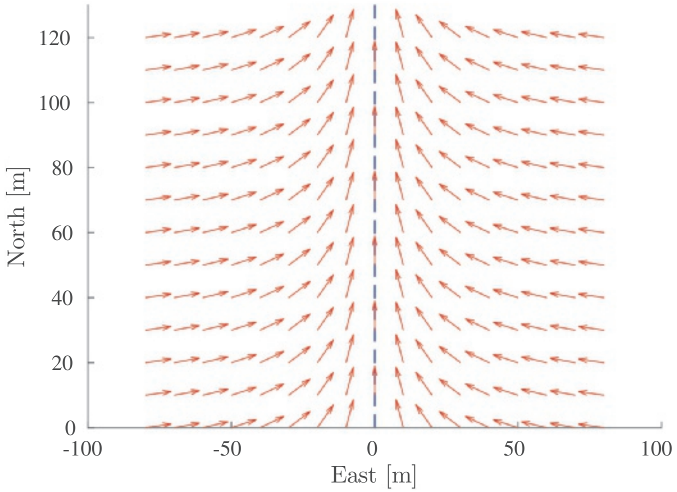


Figure 5.14: Velocity vector field corresponding to an arbitrary line.

5.6. Test flight results

During the development of the controller discussed in this chapter, many test flights have been performed. This section will present some of these test flights to support claims made throughout this chapter.

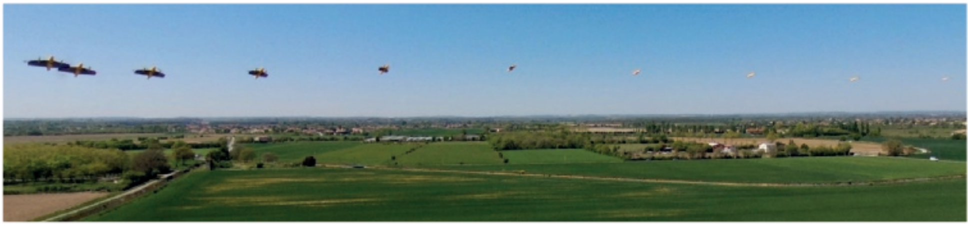


Figure 5.15: Composite image of a transition to forward flight with a constant acceleration of 1 m/s^2, with 1 s image intervals. Footage was taken with a Bebop, which was hovering in place and rotating to keep the drone in the frame.

Figure 5.15 shows a picture of a transition to forward flight, where the Cyclone was specifically commanded to constantly accelerate with 1 m/s^2. The picture is constructed from video frames, taken by a hovering bebop, that rotated in order to keep the Cyclone in the frame. The frames are taken with 1 second intervals, and stitched afterwards. The figure also gives a visual impression of the Cyclone's ability to transition while aligning itself with the direction of movement. A transition back to hover is shown in Figure 5.16, again with 1 second intervals.

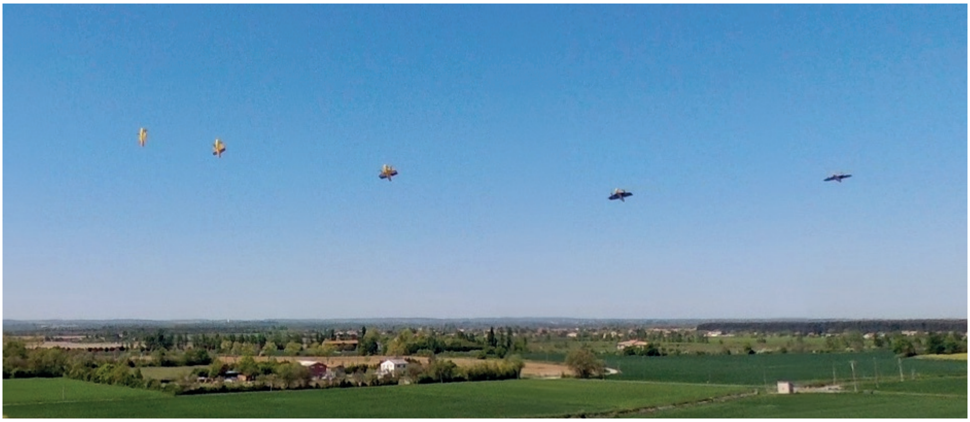


Figure 5.16: Composite image of a transition to hover flight, with 1 s image intervals. Footage was taken with a hovering Bebop.

5.6.1. Attitude control performance

First, the attitude control is shown for the case where the Cyclone is flying back and forth between two waypoints and hovers for a while at each waypoint. The waypoints are more than 200 m apart in the east direction, and the west waypoint is 17 m more northern.

From Figure 5.17 it can be seen that the pitch angle can be tracked across the flight envelope. Figure 5.19 shows the actuator inputs that were given during the flight. Even though saturation of the flaps can be observed whenever the aircraft is pitching up at high angle of attack, the pitch angle remains tracked. The shaded areas in the figure represent times at which the controller used thrust in order to provide extra pitch moment. This typically happens at the intermediate pitch angles.

Figure 5.18 shows the roll angle (ZXY Euler convention) for the same flight. When saturation of the flaps occurs, the roll is not well controlled anymore. The reason is that the pitch axis has priority over the roll axis in the case of saturation. Though the roll error is not desired, it does not lead to instability.

Section 5.4 describes how the Cyclone controls accelerations. Here, it is investigated how well the accelerations are tracked for the flight between waypoints described above. Figure 5.20 shows the accelerations in the North and East axes, along with the reference acceleration that should be tracked. From the figure it becomes clear that most of the time the accelerations are well tracked. At some instances, the acceleration is not tracked well, such as at t=606 and t=610. This can be attributed to saturation of the flaps, which prohibits complete realization of the control objective. This results in temporary loss of tracking, which is also observed for the roll in Figure 5.18, and which is restored once the actuators are not saturated any more.

A top view of the flight is shown in Figure 5.21. It shows the track, along with the airspeed vector every four seconds in red. For parts of the flight where the

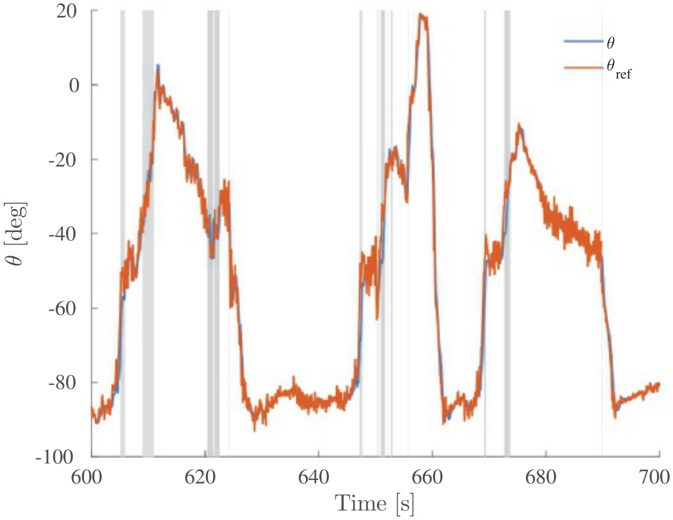


Figure 5.17: Pitch angle for the experiment (ZXY Euler). Shaded areas indicate utilization of thrust in order to pitch.

airspeed is lower than 10 m/s, the airspeed measurement is not accurate. For these vectors it is assumed that the airspeed was as large as the average estimated wind speed of 6.7 m/s, and these vectors are displayed in orange. The wind came from approximately -70 degrees north. When going to the eastern waypoint, the vehicle rotates more than 120 degrees while departing for the other waypoint. In these flights, the Cyclone did not have a certain path defined, which is why not all routes between the two waypoints exactly coincide.

5.6.2. Forward flight

To demonstrate the forward flight capabilities, Figure 5.22 shows a ground track where the vehicle consecutively follows lines, such that it tracks a polygon. The maximum airspeed was set to 16 m/s, while the desired speed along the line was 26 m/s. In the figure, the small circles represent the points at which the vehicle switches to tracking the next line. The figure demonstrates the ability to converge to and accurately track line segments. This flight was performed on a day with hardly any wind.

Figure 5.23 shows the ground track for a similar polygon on a windy day. The average wind speed was estimated at 8.3 m/s from west-southwest direction, based on the airspeed readings during the flight. The figure shows that regardless of the wind, the lines can be tracked accurately. However, the figure also shows the limitations of this straightforward method. The turn in the southeast corner structurally overshoots the line, which can be understood by noting that because of the wind, the ground speed is very high at this point. The tight turn requires accelerations that are simply not feasible. This shows the need for a proper path following algorithm, that will always respect the acceleration limitations of the vehicle.

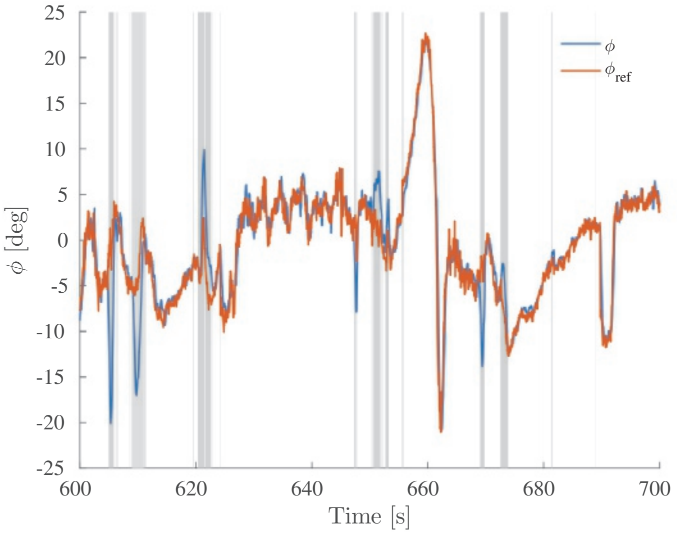


Figure 5.18: Roll angle for the experiment (ZXY Euler). Shaded areas indicate saturation of at least one of the flaps.

5.7. Guidelines for implementing INDI for hybrids

The proposed approach to attitude control, velocity control, and guidance is applicable to a wide range of hybrid UAVs. In order to facilitate the implementation of the approach on other vehicles, here, we summarize the basic steps that are needed to start flying a hybrid UAV with the INDI controller. It is assumed that it is already possible to fly the vehicle vertically, possibly by performing safe tests where the drone is attached to a rope. Relating to the sections of this chapter, the general steps can be listed as follows:

1. Identify actuator dynamics, by measuring the response over time to step inputs. For motors, measure the RPM as a function of time. For servos, the position can be obtained as a voltage from the internal potentiometer.

2. Choose a filter cutoff for the gyroscope and accelerometer noise level in flight. More filtering means that less noise is propagated to the actuators, but it also means that the system reacts slower to disturbances.

3. Identify the control effectiveness of the actuators using test flights. This can be done by fitting changes in measured data, i.e. (rotational) acceleration, with a function of changes in the control inputs and important state variables, such as airspeed.

4. Design the control gains K_Ω and K_η (Section 5.2.2) such that the attitude response is fast and stable, and repeat the previous step for forward flight.

5. Add thrust effectiveness on pitch when the flaps are saturated (if needed), by

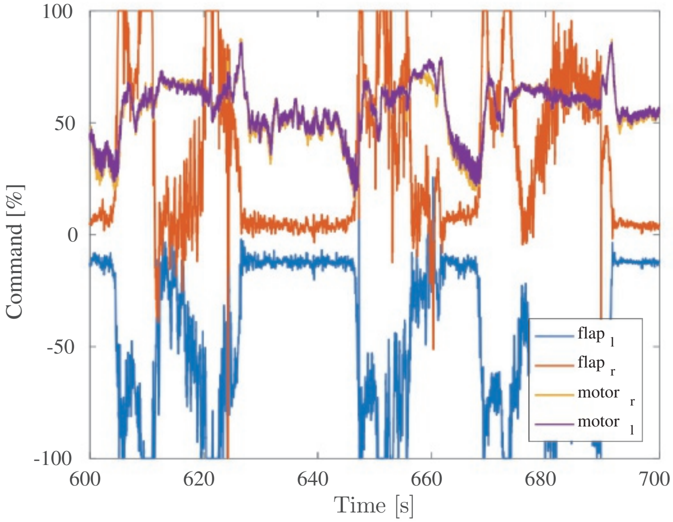


Figure 5.19: Inputs to flaps an motors during the experiment. Left flap positive down, right flap positive up.

estimating the effectiveness of the thrust on the angular acceleration in the pitch axis for large flap deflections from flight data.

6. Identify the effectiveness of changes in pitch angle on the vertical acceleration. Repeat this step for a number of airspeeds in the flight envelope, such that you can estimate a function like Equation 5.29.

7. Add effectiveness of flap deflection on the lift production, as explained in Section 5.4.1.

8. Test INDI acceleration control, and subsequently fully autonomous flight.

Steps three, six and seven should be repeated for several points in the flight envelope. This is indicated schematically in Figure 5.24. The figure also indicated which steps are part of the attitude control, position control and autonomous flight.

We have argued that INDI is an approach that does not require a lot of modelling, but still there are a few parameter estimation steps in the list above. One may wonder if it is truly less involved than a model based approach.

In the list above, note that the only thing we are estimating are control derivatives and actuator dynamics. To make a full model of the vehicle dynamics, next to the control derivatives, we would need to know the lift, drag, and moments as a function of the vehicle state and actuator inputs. This would be a lot more difficult than the simple functions for the control effectiveness we have used in this chapter. Further, the vehicle state vector necessary to make such a model accurate, must contain information about the aerodynamics, at least the angle of attack and the airspeed. These things are hard to measure at low airspeed, making it difficult to apply the model in real life.

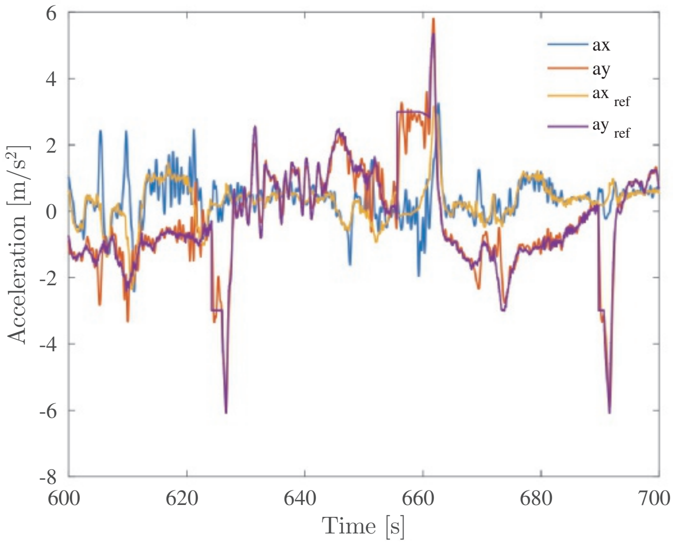


Figure 5.20: Acceleration in the North (x) and East (y) axes.

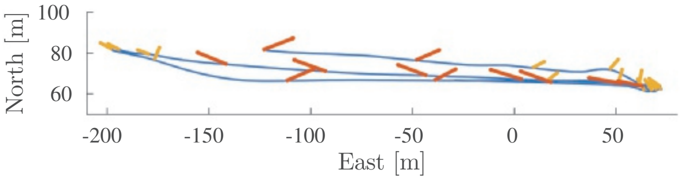


Figure 5.21: Top view of the experiment. Red arrows indicate measured airspeed, assuming no sideslip. Orange arrows indicate the estimated airspeed for low speeds.

5.8. Conclusion

We have described attitude and position control using INDI for a hybrid tailsitter UAV. Test flights show that unmodeled forces and moments, such as the strong pitch-down moment during transitions, are effectively counteracted by the incremental control structure. The resultant controller is able to track a three dimensional acceleration reference, and in doing so can autonomously transition to forward flight and back to hover. The control effectiveness is adjusted in flight to cope with the changing flight conditions, but the attitude and velocity control structure always remain the same. Only crude modeling of the control effectiveness is needed, as the incremental, sensor based approach of INDI can correct for many of the modeling errors.

Future work

We have presented ways of prioritizing the pitching moment over other control objectives, in order to be able to control the pitch angle at high angle of attack. Future work may focus on ways of increasing the actuator effectiveness, or reducing unde-

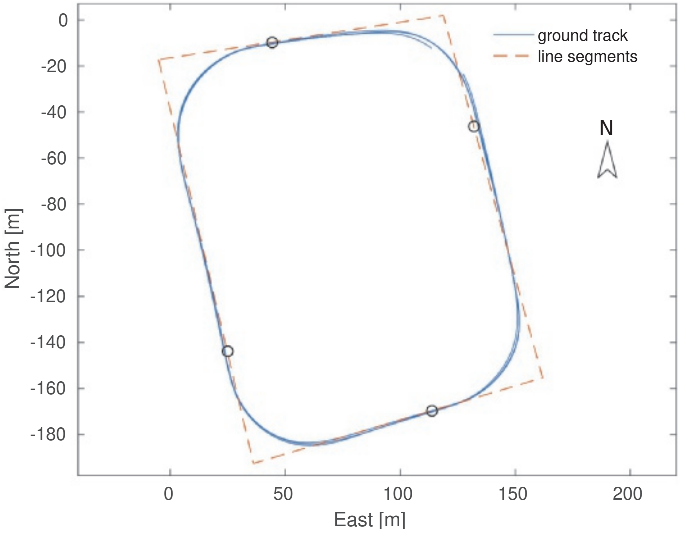


Figure 5.22: Top view of line following, the aircraft flew counter-clockwise. The line switch distance is indicated with small circles. Very weak wind.

sired moments, such that actuator saturation does not occur. If actuator saturation can be prevented, all degrees of freedom can remain controlled. Possibilities of achieving this include adding a rotor to the tail, making the rotors able to tilt, and changing the flap design.

Another topic for future research is an in-depth study into the best method of dealing with the non-minimum phase system of flap effectiveness on the lift. Further, it would be interesting to investigate the effect of each of the assumptions and simplifications.

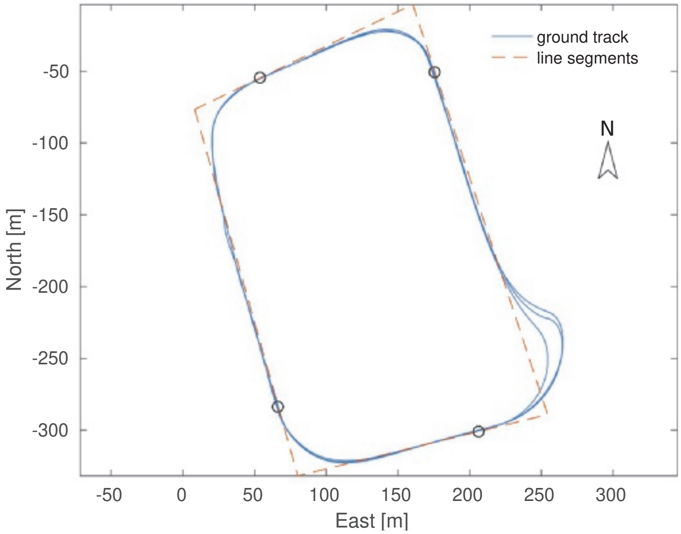


Figure 5.23: Top view of line following, the aircraft flew counter-clockwise. Considerable wind from the west. The line switch distance is indicated with small circles.

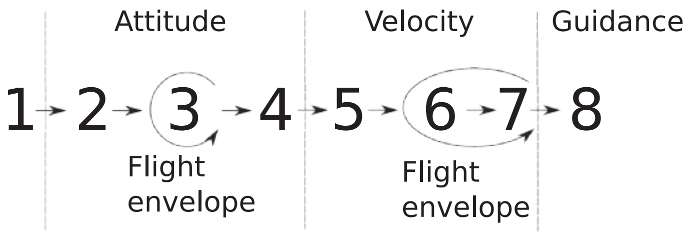


Figure 5.24: Schematic overview of the steps that need to be taken to apply INDI to a new hybrid vehicle.

6 Conclusion

In this chapter, I briefly revisit the research questions I set out to answer, in the light of the findings of Chapters 2 - 5. I subsequently provide a discussion of the contributions of this thesis. The chapter ends with an outlook on future work.

6.1. Research Questions

Recall the first research question:

> **Research question 1**
>
> How can the attitude of an unmanned aircraft be controlled with incremental nonlinear dynamic inversion?

In Chapter 2, the main finding is that the input increment loop should contain the same delays as in the angular acceleration feedback loop. It is shown, in theory and in practice, that including the same filter as is applied on the angular acceleration in the input increment loop, removes its effects from the transfer function of the complete system. It is then shown that the transfer function of the complete system is equal to the transfer function of the actuators. This result can subsequently be used to design an attitude controller using LTI systems theory. Additionally, the inertia of propellers, which causes the derivative of the input to have an effect on the angular acceleration, is dealt with by introducing an additional term in the control effectiveness inversion. It is also shown that the control effectiveness can be estimated online, in order to deal with changes in the vehicle configuration.

The second research question was formulated as:

> **Research question 2**
>
> How can the algorithm deal with control priorities in the case of actuator saturation?

In Chapter 3, the definition of priorities on different axes is formulated as a quadratic programming problem. Weights can be chosen for the different axes, as well as for the individual inputs, to define the relative priority. The active set method is then used to solve this Weighted Least Squares (WLS) programming problem and find the optimal solution, considering the constraints on the inputs. Applied to a quadrotor, with this approach it is possible to prioritize pitch and roll over yaw and thrust, greatly improving stability in the case of high desired yaw moments.

The third research question was posed as:

Research question 3

How can the velocity of an unmanned aircraft be controlled with incremental nonlinear dynamic inversion?

In Chapter 4, the INDI theory is extended to the control of linear accelerations. In a windtunnel experiment, it is shown that the INDI controller has a maximum error of 0.21 m when being hit by a gust of 10 m/s, which is 7 times better than a comparable PID controller. This shows that the disturbance rejection properties that were proven for the inner loop carry over to the outer loop. As the acceleration is controlled by the INDI controller, the velocity can be controlled with a simple proportional controller.

Finally, the last research question was:

Research question 4

How should incremental nonlinear dynamic inversion be adapted to apply it to hybrid unmanned air vehicles?

In Chapter 5, INDI is applied to the attitude and velocity control of a tailsitter hybrid UAV. For the inner loop, the control effectiveness is defined as a function of the airspeed and pitch angle in order to deal with the changes of the control effectiveness for different flight conditions. Test flights show that during the transition, large changes in inputs are required to keep tracking the pitch reference. This shows the ability of the controller to compensate large unmodeled moments in a very short time. The WLS optimization is used to prioritize pitch over roll, as the vehicle can otherwise not provide enough pitch moment to come back to hover from forward flight.

For the velocity control of quadrotor vehicles, the heading of the vehicle is not relevant and can therefore be commanded separately. This is not the case for hybrid aircraft, where the wing should provide lift. The wing can only provide lift effectively if the sideslip angle is small, constraining the heading angle. To this end, the heading angle is commanded as to minimize the sideslip angle, through a feedforward term depending on the bank angle and feedback from the accelerometer.

For the INDI acceleration controller, ZXY Euler angles are used to avoid gimbal lock problems at 90 degrees pitch angle. The ZXY Euler angles also lead to comprehensive equations for the control effectiveness of the lift of the wing on the linear acceleration. These are added to the control effectiveness of the thrust vector, producing one control effectiveness matrix. The INDI controller uses this control effectiveness matrix to track linear accelerations across the flight envelope, even though some terms are not modeled, such as the drag. Flight tests confirm that linear accelerations can be accurately tracked at each stage of the flight, except when this is physically not possible due to actuator saturation.

6.2. Discussion

When I started this research, INDI had been discussed in several papers as a method of angular acceleration control, and had been studied in simulations, but it had never been applied to a real system. In the literature, the added value of the method was presented in its ability to control nonlinear systems, which may be nonaffine in the control inputs, without much model information.

This thesis sheds new light on what is possible in practice with INDI. One of the findings in Chapter 2 was that unmodeled dynamics and disturbances are compensated through the same mechanism, and with the same transfer function. This means that next to being able to quickly compensate unmodeled effects, disturbances can also be rejected very fast. The transfer function that was found shows that this compensation is faster if the actuator dynamics are faster, and if a filter with little phase lag is used. This insight can help in improving the performance of systems with fast dynamics, or that need very fast rejection of disturbances, which can both be the case for hybrid vehicles.

In Chapter 5, it was demonstrated that it is possible to use INDI for the attitude control of a tailsitter MAV. It was observed that for high angle of attack flight, much larger flap deflections were necessary than for hover or forward flight. When transitioning from hover to forward flight and back, these large changes in flap deflection were made rapidly, while tracking the pitch angle. This shows the ability of INDI to deal with nonlinear dynamics through quick increments in the control input.

Some hybrid MAVs have poor control authority, for example the Cyclone discussed in Chapter 5, which relies on flaps inside the propeller wake for pitch and yaw moments. The WLS method provides the optimal actuator commands, if the fulfillment of the control objectives is formulated as a quadratic cost function. This is important to make sure that the vehicle has the right control priorities, but it is not a complete solution to actuator saturation. Saturation may still cause errors in the control objective, simply because the vehicle is physically unable to achieve the control objective. This can only be remedied by improving the control authority of the vehicle, by adding actuators or making existing actuators more effective.

Chapter 4 proves that it is possible to use INDI for the control of the linear accelerations of a quadrotor. It is shown experimentally that the disturbance rejection properties extend to the control of linear accelerations. The controller can compensate the vehicle drag, without any model information. The method is extended to tailsitters in Chapter 5, just by adding control derivatives for the lift of the wing. The incremental formulation leads to a very flexible control law that, together with the sideslip control, is able to exploit the entire flight envelope. There is no explicit transition maneuver programmed in the algorithm, but when the vehicle is commanded to keep accelerating, the increments in control inputs naturally lead to a transition to forward flight. The controller can be seen as an abstraction layer, that hides the complex dynamics of the vehicle from the velocity controller. This way, the velocity controller can be very simple as well, as it can simply output a desired acceleration to be tracked by the INDI controller, irrespective of the current state.

Although I have not applied the method to other types of hybrid aircraft, with the right definition of the control effectiveness this should be straightforward to accomplish. Moreover, the use of the controller presented in this thesis could lead to design simplifications for several aircraft. Most notably, we have shown in Chapter 5, that the need for a vertical surface can be removed by introducing active sideslip control based on the accelerometer. Removing the vertical surface could save weight and reduce drag. Finally, vehicles with redundant actuators, such as quadplanes, will benefit from the possibility of defining the relative priority of the actuators in the WLS algorithm.

6.3. Future Work

The research presented in this thesis can be taken further in three general directions: general INDI theory, control of hybrid MAVs, and the design of hybrids.

Regarding fundamental INDI theory, it would be very useful to have more insight in the limitations of INDI. It could be that there are systems which are so unstable, that the increments are not fast enough to stabilize the system. As the disturbance rejection transfer function is known, it may be possible, if the system under consideration is known, to find a closed form equation that determines if the controlled system is stable or not.

For systems that can be controlled with INDI, an effective method of dealing with actuator modeling errors, is using a too high control effectiveness in the controller. As the controller takes the inverse of the control effectiveness to calculate the control input, this leads to smaller, more conservative increments. These increments may not lead to the desired output, but this can be compensated later with additional increments. This gave good results for the problem of the flaps affecting the lift discussed in Chapter 5, and reasonable results for the problem of rotor inertia discussed in Chapter 2. However, a strong mathematical basis for this approach is lacking. Analysis of this method could lead to more knowledge on how and when this method should be used.

Regarding the control of hybrids, the tilt-twist attitude error formulation could be used instead of the standard quaternion error in the attitude control. This could improve the attitude tracking in cases of large yaw errors, where tilt angles could be given priority over the twist angle.

Furthermore, the guidance employed in Chapter 5 is limited to the tracking of lines, and does not take acceleration limits of the vehicle into account. This can cause the vehicle to overshoot some turns, especially with tailwind. This way, a turn that is normally feasible, can be infeasible in a windy situation. It should be investigated if the path should be replanned during flight, or if a solution could be to adjust the speed of the vehicle.

Moreover, the developed INDI controller could be applied to different hybrid vehicles. This thesis has demonstrated the potential of INDI for hybrid vehicles. It could be investigated if application to different hybrid vehicles reduces modeling costs, cuts development time and improves both disturbance rejection and tracking performance for those vehicles.

Regarding the design of hybrids, Chapter 5 has given insight in the challenges of a tailsitter with only two fixed pitch propellers. The design of the Cyclone used for this research, could be improved by increasing the pitch-up moment that the actuators can create at a high angle of attack flight. This would reduce the saturation of the flaps and as such improve the control of objectives that have low priority, such as the roll angle. Further, improving tracking of the pitch angle would lead to improved tracking of the altitude during transitions.

Acknowledgements

Though this thesis is the result of four years work at the MAVLab of the Control and Simulation (C&S) department of the Aerospace faculty of Delft University of Technology, I already joined this department as a master student back in 2011. Over the years I got to know the department well, and I am honoured to have been part of it. My thanks goes out to all the people working and studying here, who make C&S such a special place.

First I would like to thank my co-promotor and supervisor, Guido de Croon. Your advice on the research, data analysis and writing has lifted the contents of this thesis to higher level. Even when I had existential questions about the scientific process, publishing, or other things, you were there to answer them, and provide advice with my academic interests in mind. Even though I sometimes did not follow your advice, it was always very much valued.

I also want to thank my promotor, Jacco Hoekstra, for his efforts in reviewing the thesis. Our progress meetings have helped me tremendously to keep on track and focus on finishing in time. Thanks also for our informal talks, which were always inspiring.

I sincerely thank Gerrit Blaquière and the Delphi Consortium, for making this thesis possible. I really appreciate the meetings that we had together, and the sponsor meetings that I could be a part of. I always felt very welcome, and it was very interesting to experience a different research field.

Thanks also to Qiping Chu, who introduced me to INDI and encouraged me to work on it. Clearly, your advice has been a major influence in the creation of this thesis. I am really grateful for your help, interest, our discussions and the confidence that you had in me.

To Murat Bronz, who supervised me during my stay at ENAC in Toulouse in my last year, I would like to express a big thank you. Your enthusiasm, curiosity and endless energy really motivated me when I needed it. I really appreciate your drive to search for the most elegant aircraft design, and to not settle for less. But most I admire your ability to shrug and laugh when a silly mistake of mine destroys a cherished airplane.

Though I did not initially set out to do a PhD, it was the spirit of the people in the MAVLab that made me stay. It was a blessing to work in a lab where every crazy idea is met with an enthusiastic ”let’s do it!”. The fun of doing practical things with drones was a great counterbalance to writing papers. It is great to see theory come alive in the contented humming of a maneuvering drone, or see it debunked as your precious machine comes crashing down.

None of that would have been possible without the wonderful people I met and had the chance to work with in the MAVLab. Bart Remes, who never stops coming up with crazy ideas and is always looking for the fun in the work he does. Christophe de Wagter, from whom you can get an interesting disquisition on almost any subject. Erik van der Horst, who's computer magic and dedication to the students is truly unparalleled. And finally Guido de Croon, who tirelessly tries to uphold the academic standards, while still having the energy to make silly jokes.

But I owe just as much to the interesting colleagues, students and interns that worked in the lab and made it an interesting place. Among others, I would like to thank Freek, 'space' Kevin, 'flap' Kevin, Sjoerd, Andries, Roland, Kirk, Matěj, Elisabeth, Bart, Kimberly, Diana, Hann Woei, Shuo, Mario and Tom. Thanks for the fun, the help, the trips to Valkenburg, the AIVD kerstpuzzel, the ideas, the conferences, the beers, the conversations and the silly jokes. Thanks for the IMAVs, where we were programming until deep in the night, but did not forget to have fun as well, with a barbecue, a smelly swimming pool or nightly flying. Also thanks to our colleagues from the research aircraft, who were never too shy about commenting on our efforts to make airworthy drones from tape and superglue.

Furthermore, I would like to thank my fellow PhD students: Jaime, Dirk, Yazdi, Sophie, Isabel, Ivan, Kasper, Ye Zhou, Yingzhi, Ye Zhang, Sihao, Dyah, Henry, Jan, Tommaso, Tao, Jerom, Julia, Rolf, Annemarie, Ezgi, Emmanuel, Daniel. Thanks for the coffee breaks, the vrijmibos, the PhD drinks, the sailing, and the C&S barbecues. I really enjoyed the trips we made together, to Le Bourget, Dirks wedding and the awesome skiing holiday in Austria. Thanks also to the people who made my time in Toulouse into a very fun time: Hector, Elgiz, Xavier, Gautier, Gudio, Jean-Philippe, Jacson, Alexandre, Michel and Christophe. Thanks to Rick, for our endless discussions and for making my time at the Mijnbouwstraat that much more fun.

Finally, I want to thank my mother, father and sister, who were always there to support and help me in any way possible. I am grateful to be your son and brother, and to receive your love.

References

[1] R. Sambell, S. Al-Mahrooqi, C. Matheny, S. Al-Abri, and S. Al-Yarubi, *Land seismic super-crew unlocks the ara carbonate play of the southern oman salt basin with wide azimuth survey,* First Break **28**, 61 (2010).

[2] Engineering and the Environment, University of Southampton, *NI MyRIO Autonomous Quadrotor,* http://uosdesign.org/designshow2015/ni-myrio-autonomous-quadrotor/, accessed: 14-03-2018.

[3] S. Verling, B. Weibel, M. Boosfeld, K. Alexis, M. Burri, and R. Siegwart, *Full attitude control of a VTOL tailsitter UAV,* in *IEEE International Conference on Robotics and Automation (ICRA)* (©IEEE, 2016) pp. 3006–3012.

[4] *Jump 20 VTOL unmanned aerial vehicle,* https://www.naval-technology.com/projects/jump-20-vtol-unmanned-aerial-vehicle/, accessed: 19-02-2018.

[5] L. Yu, D. Zhang, and J. Zhang, *Transition flight modeling and control of a novel tilt tri-rotor UAV,* in *International Conference on Information and Automation (ICIA)* (©IEEE, 2017) pp. 983–988.

[6] P. Hartmann, M. Schütt, and D. Moormann, *Control of departure and approach maneuvers of tiltwing VTOL aircraft,* in *Guidance, Navigation, and Control Conference, AIAA 2017-1914* (2017).

[7] C. D. Wagter, R. Ruijsink, E. J. J. Smeur, K. G. van Hecke, F. van Tienen, E. van der Horst, and B. D. W. Remes, *Design, control, and visual navigation of the DelftaCopter VTOL tail-sitter UAV,* Journal of Field Robotics (2018), 10.1002/rob.21789.

[8] L. Lustosa, F. Defay, and J.-M. Moschetta, *Development of the flight model of a tilt-body MAV,* in *International Micro Air Vehicle Conference and Competition (IMAV)* (2014) pp. 157–163.

[9] B. Theys, C. Notteboom, M. Hochstenbach, and J. De Schutter, *Design and control of an unmanned aerial vehicle for autonomous parcel delivery with transition from vertical take-off to forward flight,* International Journal of Micro Air Vehicles **7**, 395 (2015).

[10] S. H. Mathisen, T. I. Fossen, and T. A. Johansen, *Non-linear model predictive control for guidance of a fixed-wing UAV in precision deep stall landing,* in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)* (2015) pp. 356–365.

[11] A. Barr and T. Greenwald, *Google working on new drone after 'wing' design failed,* The Wall Street Journal (2015).

[12] J. M. Beach, M. E. Argyle, T. W. McLain, R. W. Beard, and S. Morris, *Tail-sitter attitude control using resolved tilt-twist,* in *International Conference on Unmanned Aircraft Systems (ICUAS)* (2014) pp. 768–779.

[13] D. Chu, J. Sprinkle, R. Randall, and S. Shkarayev, *Simulator development for transition flight dynamics of a VTOL MAV,* International Journal of Micro Air Vehicles **2**, 69 (2010).

[14] P.-R. Bilodeau and F. Wong, *Modeling and control of a hovering mini tail-sitter,* International Journal of Micro Air Vehicles **2**, 211 (2010).

[15] J. L. Forshaw, V. J. Lappas, and P. Briggs, *Transitional control architecture and methodology for a twin rotor tailsitter,* Journal of Guidance, Control, and Dynamics **37**, 1289 (2014).

[16] A. Frank, J. S. McGrew, M. Valenti, D. Levine, and J. P. How, *Hover, transition, and level flight control design for a single-propeller indoor airplane,* in *Guidance, Navigation and Control Conference and Exhibit* (AIAA 2007-6318, 2007) pp. 100–117.

[17] E. Small, E. Fresk, G. Andrikopoulos, and G. Nikolakopoulos, *Modelling and control of a tilt-wing unmanned aerial vehicle,* in *Mediterranean Conference on Control and Automation (MED)* (2016) pp. 1254–1259.

[18] W. E. Green and P. Y. Oh, *A hybrid MAV for ingress and egress of urban environments,* IEEE Transactions on Robotics **25**, 253 (2009).

[19] N. B. F. Silva, J. V. C. Fontes, R. S. Inoue, and K. R. L. J. C. Branco, *Development of a fixed-wing vertical takeoff and landing aircraft as an autonomous vehicle,* in *Latin American Robotics Symposium (LARS) and Brazilian Symposium on Robotics (SBR)* (2017) pp. 1–6.

[20] A. Oosedo, S. Abiko, A. Konno, and M. Uchiyama, *Optimal transition from hovering to level-flight of a quadrotor tail-sitter UAV,* Autonomous Robots **41**, 1143 (2017).

[21] P. Hartmann, C. Meyer, and D. Moormann, *Unified velocity control and flight state transition of unmanned tilt-wing aircraft,* Journal of Guidance, Control, and Dynamics **40**, 1348 (2017).

[22] Y. Wang, X. Lyu, H. Gu, S. Shen, Z. Li, and F. Zhang, *Design, implementation and verification of a quadrotor tail-sitter VTOL UAV,* in *International Conference on Unmanned Aircraft Systems (ICUAS)* (IEEE, 2017) pp. 462–471.

[23] L. R. Lustosa, F. Defay, and J.-M. Moschetta, *Longitudinal study of a tilt-body vehicle: modeling, control and stability analysis,* in *International Conference on Unmanned Aircraft Systems (ICUAS)* (IEEE, 2015) pp. 816–824.

[24] E. N. Johnson, A. Wu, J. C. Neidhoefer, S. K. Kannan, and M. A. Turbe, *Flight-test results of autonomous airplane transitions between steady-level and hovering flight,* Journal of Guidance, Control, and Dynamics **31**, 358 (2008).

[25] N. B. Knoebel, S. R. Osborne, D. O. Snyder, T. W. McLain, R. W. Beard, and A. M. Eldredge, *Preliminary Modeling, Control, and Trajectory Design for Miniature Autonomous Tailsitters,* in *AIAA Guidance, Navigation, and Control Conference and Exhibit* (2006).

[26] Y. Jung and D. H. Shim, *Development and application of controller for transition flight of tail-sitter UAV,* Journal of Intelligent and Robotic Systems **65**, 137 (2012).

[27] W. Jin, S. Bifeng, W. Liguang, and T. Wei, *L1 adaptive dynamic inversion controller for an x-wing tail-sitter MAV in hover flight,* Procedia Engineering **99**, 969 (2015).

[28] D. Yeo, V. Hrishikeshavan, and I. Chopra, *Gust detection and mitigation on a quad rotor biplane,* in *AIAA Atmospheric Flight Mechanics Conference* (2016).

[29] C. D. Wagter, D. Dokter, G. de Croon, and B. Remes, *Multi-lifting-device UAV autonomous flight at any transition percentage,* in *Proceeding of: EuroGNC 2013* (2013) pp. 18–33.

[30] H. Gu, X. Lyu, Z. Li, S. Shen, and F. Zhang, *Development and experimental verification of a hybrid vertical take-off and landing (VTOL) unmanned aerial vehicle(UAV),* in *International Conference on Unmanned Aircraft Systems (ICUAS)* (IEEE, 2017) pp. 160–169.

[31] B. Yuksek, A. Vuruskan, U. Ozdemir, M. A. Yukselen, and G. Inalhan, *Transition flight modeling of a fixed-wing VTOL UAV,* Journal of Intelligent & Robotic Systems **84**, 83 (2016).

[32] R. H. Stone, P. Anderson, C. Hutchison, A. Tsai, P. Gibbens, and K. C. Wong, *Flight testing of the t-wing tail-sitter unmanned air vehicle,* Journal of Aircraft **45**, 673 (2008).

[33] P. R. Smith, *A simplified approach to nonlinear dynamic inversion based flight control,* in *AIAA Atmospheric Flight Mechanics Conference and Exhibit* (1998) pp. 762–770.

[34] B. J. Bacon, A. J. Ostroff, and S. M. Joshi, *Reconfigurable NDI Controller Using Inertial Sensor Failure Detection & Isolation,* IEEE Transactions On Aerospace And Electronic Systems **37**, 1373 (2001).

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

[36] R. Mahony, V. Kumar, and P. Corke, *Multirotor Aerial Vehicles: Modeling, Estimation and Control of Quadrotor,* IEEE Robotics & Automation Magazine **19**, 20 (2012).

[37] E. Fresk and G. Nikolakopoulos, *Full Quaternion Based Attitude Control for a Quadrotor,* in *European Control Conference* (IEEE, 2013) pp. 3864–3869.

[38] R. da Costa, Q. Chu, and J. Mulder, *Reentry Flight Controller Design Using Nonlinear Dynamic Inversion,* Journal of Spacecraft and Rockets **40**, 64 (2003).

[39] B. J. Bacon and A. J. Ostroff, *Reconfigurable Flight Control Using Nonlinear Dynamic Special Accelerometer Implementation,* in *Guidance, Navigation and Control Conference, Denver, CO* (AIAA, 2000).

[40] T. H. Cox and M. C. Cotting, *A Generic Inner-Loop Control Law Structure for Six-Degree-of-Freedom Conceptual Aircraft Design,* in *AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV* (2005).

[41] A. J. Ostroff and B. J. Bacon, *Enhanced NDI Strategies For Reconfigurable Flight Control,* in *Proceedings of the American Control Conference* (2002) pp. 3631–3636.

[42] J. Koschorke, W. Falkena, E.-J. van Kampen, and Q. P. Chu, *Time Delayed Incremental Nonlinear Control,* in *Guidance, Navigation, and Control Conference* (AIAA, 2013).

[43] G. D. B. Theys, T. Andrianne, P. Hendrick, and J. D. Schutter, *Wind Tunnel Testing of a VTOL MAV Propeller in Tilted Operating Mode,* in *International Conference on Unmanned Aircraft Systems* (IEEE, 2014) pp. 1064–1072.

[44] G. Chowdhary, E. N. Johnson, R. Chandramohan, M. S. Kimbrell, and A. Calise, *Guidance and Control of Airplanes Under Actuator Failures and Severe Structural Damage,* Journal of Guidance Control and Dynamics **36**, 1093 (2013).

[45] A. Bedford and W. Fowler, *Engineering Mechanics Dynamics,* (Prentice Hall, Pearson Education South Asia Pte Ltd, ISBN 981-06-7940-8, 2008) pp. 507–515.

[46] P. Simplicio, M. Pavel, E. van Kampen, and Q. Chu, *An acceleration measurements-based approach for helicopter nonlinear flight control using Incremental Nonlinear Dynamic Inversion,* Control Engineering Practice **21**, 1065 (2013).

[47] S. Haykin and B. Widrow, *Least-Mean-Square Adaptive Filters,* (Wiley, ISBN: 978-0-471-21570-7, 2003) pp. 1–12.

[48] J. Love, *Process Automation Handbook,* (Springer-Verlag London, 2007) pp. 173–178.

[49] M. W. Oppenheimer, D. B. Doman, and M. A. Bolender, *Control allocation for over-actuated systems,* in *Mediterranean Conference on Control and Automation* (2006).

[50] A. Chamseddine, I. Sadeghzadeh, Y. Zhang, D. Theilliol, and A. Khelassi, *Control allocation for a modified quadrotor helicopter based on reliability analysis,* in *Infotech@Aerospace Conference* (2012).

[51] W. C. Durham, *Computationally efficient control allocation,* Journal of Guidance Control and Dynamics **24**, 519 (2001).

[52] E. J. J. Smeur, Q. P. Chu, and G. C. H. E. de Croon, *Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Aerial Vehicles,* Journal of Guidance, Control, and Dynamics **39**, 450 (2016).

[53] E. J. J. Smeur, G. C. de Croon, and Q. Chu, *Gust disturbance alleviation with incremental nonlinear dynamic inversion,* in *International Conference on Intelligent Robots and Systems (IROS)* (IEEE/RSJ, 2016) pp. 5626–5631.

[54] T. A. Johansen and T. I. Fossen, *Control allocation - a survey,* Automatica **49**, 1087 (2012).

[55] O. Härkegård, *Efficient Active Set Algorithms for Solving Constrained Least Squares Problems in Aircraft Control Allocation,* in *41th IEEE Conference on Decision and Control* (2002).

[56] P. de Lamberterie, T. Perez, and A. Donaire, *A low-complexity flight controller for unmanned aircraft systems with constrained control allocation,* in *2011 Australian Control Conference* (2011) pp. 284–289.

[57] M. Faessler, D. Falanga, and D. Scaramuzza, *Thrust mixing, saturation, and body-rate control for accurate aggressive quadrotor flight,* IEEE Robotics and Automation Letters **2**, 476 (2017).

[58] M. Bodson, *Evaluation of optimization methods for control allocation,* Journal of Guidance, Control, and Dynamics **25**, 703 (2002).

[59] J. C. Monteiro, F. Lizarralde, and L. Hsu, *Optimal control allocation of quadrotor UAVs subject to actuator constraints,* in *American Control Conference* (2016).

[60] P. Sinha, P. Esden-Tempski, C. Forrette, J. Gibboney, and G. Horn, *Versatile, modular, extensible VTOL aerial platform with autonomous flight mode transitions,* in *IEEE Aerospace Conference* (2012).

[61] J. A. M. Petersen and M. Bodson, *Constrained quadratic programming techniques for control allocation,* IEEE Transactions on Control Systems Technology **14**, 91 (2006).

[62] A. ke Björck, *Numerical methods for least squares problems,* (Siam, 1996) pp. 187–213.

[63] M. Bronz, E. J. J. Smeur, H. G. de Marina, and G. Hattenberger, *Development of a fixed-wing mini UAV with transitioning flight capability,* in *AIAA Aviation Forum* (2017).

[64] A. Ryan and J. Hedrick, *A mode-switching path planner for UAV-assisted search and rescue,* in *44th IEEE Conference on Decision and Control* (2005) pp. 1471–1476.

[65] R. D'Andrea, *Can drones deliver?* IEEE Transactions on Automation Science and Engineering **11**, 647 (2014).

[66] J. Kim and S. Sukkarieh, *Airborne simultaneous localisation and map building,* in *IEEE International Conference on Robotics and Automation* (2003) pp. 406–411.

[67] K. Alexis, G. Nikolakopoulos, and A. Tzes, *Constrained-Control of a Quadrotor Helicopter for Trajectory Tracking under Wind-Gust Disturbances,* in *IEEE Mediterranean Electrotechnical Conference* (2010) pp. 1411–1416.

[68] M. W. Orr, S. J. Rasmussen, E. D. Karni, and W. B. Blake, *Framework for Developing and Evaluating MAV Control Algorithms in a Realistic Urban Setting,* in *American Control Conference* (2005) pp. 4096–4101.

[69] S. Shen, N. Michael, and V. Kumar, *Autonomous Multi-Floor Indoor Navigation with a Computationally Constrained MAV,* in *International Conference on Robotics and Automation* (2011) pp. 20–25.

[70] D. Mellinger, N. Michael, and V. Kumar, *Trajectory generation and control for precise aggressive maneuvers with quadrotors,* The International Journal of Robotics Research **31**, 664 (2012).

[71] N. Sydney, B. Smyth, and D. A. Paley, *Dynamic control of autonomous quadrotor flight in an estimated wind field,* in *IEEE Conference on Decision and Control (CDC)* (2013) pp. 3609–3616.

[72] A. Mohamed, M. Abdulrahim, S. Watkins, and R. Clothier, *Development and flight testing of a turbulence mitigation system for micro air vehicles,* Journal of Field Robotics **33**, 639 (2016).

[73] J. Escareño, S. Salazar, H. Romero, and R. Lozano, *Trajectory Control of a Quadrotor Subject to 2D Wind Disturbances,* Journal of Intelligent & Robotic Systems **70**, 51 (2012).

[74] S. L. Waslander and C. Wang, *Wind Disturbance Estimation and Rejection for Quadrotor Position Control,* in *AIAA Infotech@Aerospace Conference and AIAA Unmanned...Unlimited Conference* (2009).

[75] F. Schiano, J. Alonso-Mora, K. Rudin, P. Beardsley, R. Siegwart, and B. Siciliano, *Towards Estimation and Correction of Wind Effects on a Quadrotor UAV,* in *International Micro Air Vehicle Conference and Competition (IMAV)* (2014) pp. 134–141.

[76] T. Tomić, K. Schmid, P. Lutz, A. Mathers, and S. Haddadin, *The Flying Anemometer: Unified Estimation of Wind Velocity from Aerodynamic Power and Wrenches,* in *International Conference on Intelligent Robots and Systems (IROS)* (Daejeon, Korea, 2016) pp. 1637–1644.

[77] G. Hattenberger, M. Bronz, and M. Gorraz, *Using the Paparazzi UAV System for Scientific Research,* in *International Micro Air Vehicle Conference and Competition (IMAV)* (2014) pp. 247–252.

[78] B. Remes, D. Hensen, F. van Tienen, C. de Wagter, E. van der Horst, and G. de Croon, *Paparazzi: how to make a swarm of Parrot AR Drones fly autonomously based on GPS.* in *International Micro Air Vehicle Conference and Flight Competition (IMAV)* (Toulouse, France, 2013) pp. 307–313.

[79] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, *Precision flight control for a multi-vehicle quadrotor helicopter testbed,* Control Engineering Practice **19**, 1023 (2011).

[80] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, *Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment,* in *Guidance, Navigation and Control Conference and Exhibit* (AIAA Paper 2007-6461, 2007).

[81] J. Wang, T. Raffler, and F. Holzapfel, *Nonlinear Position Control Approaches for Quadcopters Using a Novel State Representation,* in *Guidance, Navigation and Control Conference* (AIAA Paper 2012-4913, 2012).

[82] A. Tayebi and S. McGilvray, *Attitude stabilization of a VTOL quadrotor aircraft,* IEEE Transactions on Control Systems Technology **14**, 562 (2006).

[83] T. Lee, M. Leoky, and N. H. McClamroch, *Geometric tracking control of a quadrotor UAV on SE(3),* in *49th IEEE Conference on Decision and Control* (2010) pp. 5420–5425.

[84] J. T. Y. Wen and K. Kreutz-Delgado, *The attitude control problem,* IEEE Transactions on Automatic Control **36**, 1148 (1991).

[85] A. A. Mian and W. Daobo, *Modeling and backstepping-based nonlinear control strategy for a 6 dof quadrotor helicopter,* Chinese Journal of Aeronautics **21**, 261 (2008).

[86] Y. Yu, S. Yang, M. Wang, C. Li, and Z. Li, *High performance full attitude control of a quadrotor on so(3),* in *IEEE International Conference on Robotics and Automation (ICRA)* (2015) pp. 1698–1703.

[87] E. J. J. Smeur, D. Höppener, and C. de Wagter, *Prioritized control allocation for quadrotors subject to saturation,* in *International Micro Air Vehicle Conference and Flight Competition 2017* (Toulouse, France, 2017) pp. 37–43.

[88] G. D. Francesco and M. Mattei, *Modeling and incremental nonlinear dynamic inversion control of a novel unmanned tiltrotor,* Journal of Aircraft **53**, 73 (2016).

[89] C. Chen, J. Zhang, D. Zhang, and L. Shen, *Control and flight test of a tilt-rotor unmanned aerial vehicle,* International Journal of Advanced Robotic Systems **14**, 1 (2017).

[90] R. Ritz and R. D'Andrea, *A global controller for flying wing tailsitter vehicles,* in *International Conference on Robotics and Automation (ICRA)* (IEEE, 2017) pp. 2731–2738.

[91] N. B. Knoebel and T. W. McLain, *Adaptive quaternion control of a miniature tailsitter UAV,* in *American Control Conference* (IEEE, 2008) pp. 2340–2345.

[92] E. J. J. Smeur, G. C. H. E. de Croon, and Q. P. Chu, *Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection,* Control Engineering Practice **73**, 79 (2018).

[93] L. L. M. Veldhuis, *Review of propeller-wing aerodynamic interference,* in *24th International Congress of the Aeronautical Sciences* (2004).

[94] J. Katz and A. Plotkin, *Low-speed aerodynamics,* (Cambridge University Press, 2001) pp. 106–110.

[95] J. Lv, J. Li, and D. An, *Coordinated-turn control law design of aircraft based on optimal tracking,* in *International Conference on Automatic Control and Artificial Intelligence* (IEEE, 2012) pp. 730–733.

[96] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control system design,* (Pearson, 2000) pp. 199–232.

[97] H. G. de Marina, Y. A. Kapitanyuk, M. Bronz, G. Hattenberger, and M. Cao, *Guidance algorithm for smooth trajectory tracking of a fixed wing UAV flying in wind flows,* IEEE International Conference on Robotics and Automation (ICRA) , 5740 (2017).

Curriculum Vitæ

Ewoud Jan Jacob SMEUR

25-06-1990	Born in Nijmegen, The Netherlands.

Education

2008-2011	BSc in Aerospace Engineering Delft University of Technology
2011-2014	MSc in Aerospace Engineering Delft University of Technology
2014-2018	PhD. in Aerospace Engineering Delft University of Technology *Thesis:* Incremental control of hybrid micro air vehicles *Promotor:* Prof. dr. ir. J.M. Hoekstra

Experience

2012	Internship at Transition Robotics Inc. Santa Cruz, California, USA
2017	Visiting researcher at the dronelab of ENAC Toulouse, France *Research focus* Control and guidance of the Cyclone hybrid UAV

Awards

2012	UfD-Imtech bachelor grant
2013	First prize at the International Micro Air Vehicle competition (IMAV)
2016	Second prize at the UAV Challenge in Australia
2016	Best paper at the International Micro Air Vehicle conference (IMAV)

List of Publications

Journal Articles

4. **E.J.J. Smeur**, M. Bronz, G.C.H.E. de Croon, *Incremental control and guidance of hybrid aircraft applied to the Cyclone tailsitter UAV*, [SUBMITTED] (2018).

3. **E.J.J. Smeur**, G.C.H.E. de Croon, and Q.P. Chu, *Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection*, Control Engineering Practice **73**, 79 (2018).

2. C. de Wagter, and **E.J.J. Smeur**, *Control of a hybrid helicopter with wings*, International Journal of Micro Air Vehicles **9**, 209 (2017).

1. **E.J.J. Smeur**, Q. P. Chu, and G. C. H. E. de Croon, *Adaptive incremental nonlinear dynamic inversion for attitude control of micro aerial vehicles*, Journal of Guidance, Control, and Dynamics **39**, 450 (2016).

Conference Proceedings

6. **E.J.J. Smeur**, D. Höppener, and C. De Wagter, *Prioritized control allocation for quadrotors subject to saturation*, in International Micro Air Vehicle Conference and Competition (IMAV) (2017) pp. 37-43.

5. E. van der Sman, **E.J.J. Smeur**, B. Remes, C. De Wagter, and Q.P. Chu, *Incremental nonlinear dynamic inversion and multihole pressure probes for disturbance rejection control of fixed-wing micro air vehicles*, International Micro Air Vehicle Conference and Competition (IMAV) (2017) pp. 111-120.

4. M. Bronz, **E.J.J. Smeur**, H. G. de Marina, and G. Hattenberger, *Development of a fixed-wing mini UAV with transitioning flight capability*, in AIAA Aviation Forum (2017) pp. 1-14.

3. **E.J.J. Smeur**, G. C. de Croon, and Q. Chu, *Gust disturbance alleviation with incremental nonlinear dynamic inversion*, in International Conference on Intelligent Robots and Systems (IROS) (2016) pp. 5626-5631.

2. B. D. W. Remes, P. Esden-Tempski, F. van Tienen, **E.J.J. Smeur**, C. de Wagter, and G. C. H. E. de Croon, *Lisa-s 2.8g autopilot for GPS-based flight of MAVs*, in International Micro Air Vehicle Conference and Competition (IMAV) (2014) pp. 280-285.

1. **E.J.J. Smeur**, Q.P. Chu, G.C.H.E. de Croon, B.D.W. Remes, C. de Wagter, and E. van der Horst, *Modelling of a hybrid UAV using test flight data*, in International Micro Air Vehicle Conference and Competition (IMAV) (2014) pp. 196-203.