Modelling and Control of a Vertical Take-Off and Landing Fixed-Wing Unmanned Aerial Vehicle

Torbjørn Kringeland

Master's Thesis
Master of Science in Electronics and Computer Technology - Cybernetics
30 credits

Department of Physics
Faculty of Mathematics and Natural Sciences

UNIVERSITY OF OSLO

Spring 2019

Modelling and Control of a Vertical Take-Off and Landing Fixed-Wing Unmanned Aerial Vehicle

Torbjørn Kringeland

Modelling and Control of a Vertical Take-Off and Landing Fixed-Wing Unmanned Aerial Vehicle

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

Rotorcraft and fixed-wing aircraft are generally considered separate concepts, each with inherent strengths and weaknesses. The last decade has seen an increasing interest in hybrid unmanned aerial vehicles (UAV) that combine the vertical take-off and landing (VTOL) capabilities of rotorcraft with the energy efficient prolonged flight of a fixed-wing. In the work with airborne sensory platforms the Norwegian Defence Research Establishment (FFI) has developed a concept plane named Kestrel. Kestrel is a dual system convertiplane hybrid UAV equipped with four upwards facing propellers for VTOL as well as a full fixed-wing control system. This thesis is devoted towards enabling future research on and with the Kestrel UAV.

Little systematic research exist concerning modelling, estimation and control of the dual system convertiplane design through its complete flight envelope. With basis in the design of the Kestrel a full six degrees of freedom dynamic model considering the main forces and moments acting upon the aircraft was developed.

The aircraft was simulated using different models during hovering and longitudinal fixed-wing flight. Linear parameter-varying (LPV) multiple model adaptive (MMA) methods were investigated as solution concepts estimation and control of the UAV during vastly different flight conditions under large uncertainties in the physical parameters. Linear state-observers were used for state estimation, and infinite horizon linear-quadratic regulators (LQR) with integral action, were used for control in both models.

In quadcopter mode the controller managed to track setpoints and follow trajectories accurately. Within the limitations of the linearised system the MMA algorithm was able to correctly identify the candidate model with parameters most similar to the true parameters of the plant.

During longitudinal flight the aircraft managed level flight as well as tracking trajectories in desired airspeed and altitude. The MMA algorithm was able to identify the correct candidate model during flight close to the computed trim state of the models. Flight conditions not encapsulated by the computation of the trim conditions caused other factors than the unknown parameter to determine the performance of the linearised systems, rendering the MMA algorithm unable to identify the correct candidate model.

An unsuccessful attempt was made at implementing the LPV architecture. The method, however, is still considered a good candidate for enabling estimation, parameter identification and control of the aircraft over a larger set of flight conditions. This shows great promise towards ensuring stable transition flight between quadcopter hovering and fixed-wing cruising.

Sammendrag

Helikopter og fixed-wing fly anses vanligvis som separate konsepter, hver med sine egne styrker og svakheter. De siste ti årene har det vært en økende interesse rettet mot hybrid ubemannede luftfartøy (UAV) som forsøker å kombinere vertikal take-off og landing (VTOL) fra helikoptre med flyenes mer energieffektive egenskaper over lengre flyvninger. I forbindelse med forskning på luftbårne sensorplatformer har Forsvarets Forskningsinstitutt (FFI) utviklet et konseptfly kalt Kestrel. Kestrel er et dual propulsion hybrid UAV utstyrt med fire propeller for VTOL i tillegg til et komplett kontrollsystem for fixed-wing flyvning. Denne oppgaven er laget for å tilrettelegge for videre forskning på og med Kestrel UAV.

Det eksisterer lite forskning som ser på methoder for modellering, estimering og kontroll av dual propulsion fartøy. Med Kestrel som utgangspunkt har en dynamisk modell over all seks frihetsgrader som tar for seg de viktikste kreftene og rotasjonsmomentene blitt utviklet.

Et luftfartøy har blitt simulert ved bruk av ulike modeller under sveving og langsgående fixed-wing flyinger. Lineær parameter-variende (LPV) multippel modell adaptive (MMA) metoder ble undersøkt som potensielle løsningskonsepter for estimering og kontroll av UAVen over vidt forskjellige flyforhold mens det var store usikkerheter knyttet til flyets fysiske parametere. Lineære metoder ble brukt for estimering, og en lineær kvadratisk regulator (LQR) ble brukt for kontroll i begge flymoduser.

I quadcopter modus klarte kontrollalgoritmen å følge setpunkt. Så lenge bevegelsen til UAVen var innen begrensningene til de lineære modellene klarte MMA algoritmen å identifisere korrekt kandidatmodell.

For langsgående flyvning klarte flyet stabile flyvninger, samt å følge variasjoner i setpunkt av ønsket høyde og lufthastighet. Under flyvninger nær linear-iseringspunktene klarte MMA algorithmen å identifisere korrekt kandidatmodell. Da flyet beveget seg bort fra disse linear-iseringspunktene var det andre faktorer som ble avgjørende for kvaliteten på de ulike lineære modellene. Dette gjorde MMA algoritmen ute av stand til å identifisere korrekt kandidatmodell.

En LPV akitektur ble forsøkt implementert uten hell. Til tross for dette er den fortsatt ansett som en mulig løsningsmetode for å realisere stabil transisjon

mellom svening i quadrocopter modus og flyvning i fixed-wing modus.

Contents

List of Figures

List of Tables

Preface

This is the concluding part of my two year Master of Science degree in Cybernetics at the University of Oslo (UiO). The degree was conducted from August 2017 to May 2019. In totality the degree consisted of 120 credits, whereof 30 were conducted during this thesis. The degree was conducted in collaboration between UiO and the Norwegian Defence Research Establishment (Forsvarets Forskningsinstitutt, FFI). FFI is developing a concept of mobile sensory platforms and has previously developed a dual system convertiplane hybrid unmanned aerial vehicle(UAV) named Kestrel. This work is a continuation of the work on the Kestrel UAV by building a mathematical model for the complete system, designing controllers and proposing methodologies for addressing possible future challenges. This was done in order to hopefully enable future research both with and on the Kestrel UAV.

First and foremost I would like to thank my supervisor Andrea Cristofaro who has been a tremendous help with his uncanny ability to quickly understand the challenges I have been facing and afterwards pointing me in the right direction

I would also like to thank my supervisor Gjermund Kjerkreit from FFI who participated in building the Kestrel, and gave me the opportunity to be a part of the project.

Jonas Moen is a researcher at FFI and lecturer at UiO. I would like to thank him for presenting this opportunity to me, and giving me a summer internship at FFI to conduct some preliminary research towards my thesis.

Finally I would like to thank everyone in my class at the Cybernetics study, we did it bois! Thank you for being such an awesome group of people, it has meant the world to me.

Chapter 1

Introduction

1.1 Unmanned Aerial Vehicles In a Historical Perspective

One of the things that have fascinated mankind the most throughout history is probably the ability to fly. The first known example of a self-propelled unmanned aerial vehicle (UAV) was a mechanical bird attributed to Archytas of Tarantas in 425 BC. In 1483 Da Vinci created an aircraft capable of performing vertical flight. During the 18th century several designs appeared that resemble the modern helicopter. In 1840 Horatio Phillips made a steam powered machine capable of vertical flight. All of these were good ideas, but the technology was still not at the level required to construct systems for practical use.

The first documented use of UAVs for war was in 1849 by the Austrian army during their attack on Venice in Italy, where balloons were fitted with explosives that was dropped onto the city. From there not much development happened until world war 1, in 1916 when Britain and later USA started experimenting with using unmanned fixed-wing vehicles armed with explosives intended to shoot down German zeppelins. Development continued during the interwar period and into the second world war, these were however for the most part planes re-purposed into flying bombs. It is known, without too much detail, that USA and USSR both had their own respective UAV programs during the cold war, they were however for the most part perceived as expensive and unreliable toys compared to their manned counterparts.

Practical use of rotorcrafts had a much slower start, even if many of the ancient inventions featured helicopter-like designs. This is much due to the inherent instability of multirotors which in the absence of computers demands immense work from the pilot. The first takeoff experiments with rotorcrafts were done with multirotor designs. In 1907 the French brothers Jacques and Louis Breguet built and took off with their quadcopter *Gyroplane No 1*, it proved to be unstable and impractical. In 1924 the Frenchman Étienne Oehmichen made great progress by flying 1 km, but quadcopters lost the race against single rotor helicopters which are inherently more stable and easier to craft. Several

decades and, technological breakthroughs were required before quadcopters would see practical use.

In 1982 during the Lebanon war Israel defeated the Syrian Air Force by using UAVs alongside manned aircraft as decoys, jammers and reconnaissance units. After this powerful demonstration of capabilities, research on UAVs really took off, pun intended. With the improvements of computers and electronics, small quadcopters also started to make their entry of practical use during the 90's offering a whole new range of capabilities that fixed-wing UAVs are not capable of.

In the last couple of decades the world of UAVs has seen a tremendous development. A huge increase can be seen in the use of UAVs for military purpose such as surveillance and reconnaissance. Until the early 2000's UAVs were mostly considered for military use, but the last couple of decades the civilian market has exploded and they are being used in situations ranging from disaster relief and wildfires to racing and glorified selfie-sticks [18, 33].

1.2 Hybrid UAVs

In the UAV market there are generally two main types considered: fixed-wing, and rotorcraft. Both with their own strengths and disadvantages. Fixed-wing UAVs generally exhibit higher translational speed, payload capacity, range and endurance. But they also require runways for takeoff and landing and are required to stay above a certain flight speed to stay airborne. Rotorcrafts on the other hand come in many different sizes and configurations, and generally offer higher manoeuvrability where it can navigate confined spaces and do not require specially designated takeoff and landing areas. However, lift is only generated from the propellers which requires a lot of energy, and will greatly impede the rotorcrafts ability on missions where endurance is required.

From these opposite concepts, a new type of UAVs that attempts to combine the advantages of both types has emerged, called hybrid UAVs. Hybrid UAVs aim at getting vertical take-off and landing (VTOL), and while in the air transition to and from fixed-wing cruising. The idea of combining these two concepts is by no means a novel idea. There has been several examples of manned aircraft achieving this, with maybe the most well known being Bell-Boeing V-22 Osprey which was the first tilt-rotor aircraft and is still in use today [9]. Only in the last decade or so has it been possible to realise unmanned versions [41].

1.3 The Kestrel

The are many possible designs within the area of hybrid UAVs, and there exist several examples of successful aircraft both in industry and research. There

are several commercial options available for purchase, but they are often quite expensive and come with limited options for modifications.

Due to these limitations a group of students in 2016 at the University of Southampton together with The Norwegian Defence Research Establishment (Forsvarets Forskningsinstitutt, FFI) developed the first generation of a concept plane named *Kestrel* that be seen in Figure (1.1) [16].

The reason for FFI to develop something of its own was to be able to build and modify everything in-house. That way it will be a lot cheaper to produce, and make it open for any kind of modifications to functionality and even alter the whole design concept. The ideas and concepts regarding Kestrel will be reviewed in Chapter (2)

Figure 1.1: Kestrel Mk II

The Kestrel is a *dual propulsion convertiplane* capable of both VTOL and fixed-wing flight. In essence a dual propulsion convertiplane is a fixed-wing UAV and a quadcopter merged together. The idea is that the quadcopter propellers will do the vertical take-off, and then the forward propeller for fixed-wing flight accelerate the aircraft through a transition phase until it reaches stable flight conditions for the fixed-wing mode. Quadcopters and fixed-wing UAVs are both well documented in the literature, as well as inheriting a lot from research on manned aircraft, rockets and helicopters. There exist several examples of commercially available dual propulsion convertiplanes such as Arcturus Jump or ALTI Transition [24, 25]. However, despite the relative simplicity in design and short distance from well developed areas of control research there is astonishingly little research done in analysing the complete system compared to other types of hybrid system UAVs [41]. The small amount of research that I was able to find is mostly concerned with concepts regarding the aircraft

design, and implementing a proportional-integral-derivative(PID) controller [13, 47].

1.4 Control of UAVs

The most widely used control algorithm around is the notorious proportional-integral-derivative(PID) controller. Despite its simplicity it has proven itself able to function fairly well in a wide range of applications by only comparing the measured state of the system with the desired state and based on this linearly compute an input which is sent to the plant in order to drive the system to its desired setpoint. This is done without considering the dynamics of the system [4]. This simplicity makes it a good starting point when designing a controller, and makes it revered among hobbyists as well as professionals. In the survey done by Saeed et. al.[41] they found that in the literature on hybrid UAVs the most common control method is PID control despite the often strong non-linearity of the system. There are several commercial autopilots available, the two most common are Pixhawk [32] and ArduPilot [2]. Both of which, out of the box, supports several different kinds of hybrid UAVs, including dual propulsion convertiplanes, or QuadPlane VTOL as it is conveniently named. The Pixhawk autopilot was also the basis for the control system used on the Kestrel by Holm et. al.[16] during their master thesis, and they were able to manually control their *Mk I* prototype through a transition from VTOL to fixed-wing flight. So, "just" getting an aircraft like the Kestrel to fly, should be feasible using either a Pixhawk or ArduPilot. These commercial autopilots are made for hobbyists, and thus are made as simple as possible in the sense that no advanced knowledge regarding modelling or control is needed in order to make an UAV fly. These commercial autopilots are based on PID controllers.

Hand-tuning a PID controller can be quite tedious work. Several methods exist that attempt to make guidelines in order to achieve control of the process with good system behaviour. When good tuning of the controller is achieved, it is only expected to behave nicely within reasonable ranges of flight conditions around the point which it was linearised. There is also no way of proving or guaranteeing stability of the system.

A fixed-wing UAV is a highly non-linear system, and the most common way of designing controls for such an UAV is by linearising the model around certain trim conditions. These trim conditions assume constant velocity, constant turn rate, constant attitude etc. However, in general fixed-wing air crafts are required to operate in a wide range of environments where the assumptions of linearisations no longer hold, and thus the quality of the control will quickly deteriorate. For this reason an extensive research was started up during the 50's on *adaptive control* for autopilot design for high-performance aircraft [5]. Looking up any resource on adaptive control, there exist a myriad of methods for adaptive control such as *pole placement adaptive control*, *gain*

scheduling, *model reference adaptive control* and many more[23]. There is no universally agreed upon definition of what adaptive control is, other than that a feedback controller with constant gain parameters is not enough, there must exist adjustable parameters and a mechanism for adjusting the parameters in the controller.

For Kestrel, during hover the assumptions of symmetry might not apply as they usually do for quadcopters. During the transition phase from VTOL to fixed-wing, and back, as well as during fixed-wing flight, the flight conditions will change quickly, and one single linearisation around a single trim state will not be sufficient. The linear parameter-varying (LPV) paradigm[42] is one possible way of dealing with these kinds of systems. LPV is a simple yet efficient method for analysis and control of non-linear systems and has seen successful application in a wide range of applications[11, 38, 44]. In addition, sudden changes in the aircraft structure or parameters due to requirements of some experiment with a new payload may occur. Instead of going through the steps of redoing accurate modelling of the aircraft or fine-tuning a new controller, applying an adaptive controller that would be able to dynamically adapt to changes in process dynamics and parameter uncertainties is desirable. One method is Multiple-Model Adaptive Estimation(MMAE) and Multiple-Model Adaptive Control(MMAC), which were invented to cope with dynamic systems with significant parameter uncertainties that could not be handled by standard techniques[6]. In a nutshell multiple-model methods basically assume that the unknown parameters lie within a fixed range of values, and that a finite set of models each taking on a discrete value within the parameter range can be designed. The idea is then that the algorithm shall correctly identify the model and controller with parameter values that most closely resemble the parameter values of the plant it attempts to estimate/control. MMA and LPV methods has previously been successfully applied to both quadcopters and fixed-wing aircraft for both estimation and control [26, 38, 39].

For controlling the aircraft in this thesis infinite horizon linear-quadratic regulator (LQR) controllers will be used with added integral action [46]. The LQR controller is still a linear controller in the same manner as the PID and requires a linearisation of non-linear models, but can guarantee convergence of the system. Computing the LQR feedback gain is based on the state-space matrices of the model, and only requires tuning from the user is of the model agnostic weighting matrices. This makes it possible to quickly compute controller gains for several different models using the same behavioural criterion defined in the weighting matrices. By constructing controllers for the different models of the LPV and MMA methods and letting the controller adapt together with these systems a multiple model linear parameter varying controller can be achieved.

1.5 Goals of this Thesis

With the development of an UAV there exists a myriad of problems to analyse and approaches to take. Considering that there is little, if any at all, systematic research targeted towards dual system convertiplanes, the main goal of this thesis will be to create a solution concept for modelling, estimation and control of a convertiplane. With special emphasis on the Kestrel UAV at FFI some of the expected challenges that may occur during its future development will be addressed. The thesis will offer possible solutions to these challenges as a step towards enabling further research both on and with the Kestrel UAV.

In the literature there are many examples of both VTOL and fixed-wing aircraft, but no examples that regard convertiplanes as a single system. In this thesis, the general six degrees of freedom (DoF) dynamic and kinematic equations will first be derived. Following, work will be conducted with relevant literature in order to thoroughly describe the main forces and moments affecting the dynamics of the convertiplane.

During hover and fixed-wing mode the dynamics of the aircraft are expected to look vastly different, and the aircraft will utilise different parts of the dual control system in order to maintain flight. There are many examples in the literature concerning estimation and control of either of the two systems, but none that describe a transition phase between two while ensuring stable flight. This thesis aims to simulate the aircraft both as a quadcopter model, as well as a fixed-wing aircraft model. A goal is then to develop a control architecture which enable both hovering and fixed-wing mode of flight, that can be extended to connect the two models during a transition phase. The architecture should be able to estimate and control the UAV with large uncertainties in the dynamic parameters of the UAV over a wide range of flight conditions. The control architecture that will be utilised to achieve this is a linear-quadratic regulator in combination with linear parameter-varying multiple model adaptive methods.

1.6 Outline of Thesis

The outline of the thesis is as follows.

In Chapter 2 the Kestrel will be presented in further detail. Containing both design philosophy and functionality.

Chapter 3 is the theory chapter. The kinematic and dynamics for an UAV are developed resulting in the dynamic equations that describe the full six DoF movement. After that the governing forces and moments that affect the Kestrel are investigated. The second part of the chapter is a presentation of control and estimation theory and describe the MMA and LPV algorithms.

In Chapter 4 the processes of how the simulations models and controllers that will be used in the various simulations are described.

Chapter 5 presents the achieved results from the various simulations.

Chapter 6 and 7 contain discussions, suggestions for further work and concluding remarks.

Chapter 2

The Kestrel UAV

The Kestrel UAV was not developed as part of this thesis, nor do any of the simulations feature a model of Kestrel explicitly. The thesis is, however, designed wholeheartedly towards analysing the flight properties of Kestrel and similar UAVs. For this reason this chapter will give an introduction of the Kestrel, specifications and design consideration, and compare this up against other hybrid UAV concepts.

2.1 Hybrid UAV Platforms

As mentioned in the Introduction Kestrel can be categorised as a dual-propulsion convertiplane hybrid UAV, this is by no means a strict naming convention but may be one of the more descriptive and more accurate ways to distinguish it from all the other types of aircraft capable of both vertical take-off and landing and fixed-wing flight. Often if may be found named as QuadPlane or VTOL-FW UAV. Following is a series of some of the more common designs that are present in the realm of hybrid UAVs. On the most general level hybrid UAVs can be distinguished by two main categories: *Tailsitters* and *Convertiplanes*.

2.1.0.1 Tail-sitters

Tail-sitters is a concept where UAV often look very much the same as a normal fixed-wing UAV, but starts off by sitting on its tail. It can then take off vertically by using its propeller(s) in the same manner as a rotorcraft, and then transition to cruise by tilting the whole body into a "normal" fixed-wing attitude. The first attempts at these kind of aircraft started in the 50's and today there are several successful examples, such as; The V-BAT which consist of a single ducted-fan propeller, and VertiKUL which use four propellers aligned in a quadcopter-like manned parallel with the fixed-wing airframe [3, 19]. Both of them can

be seen in Figure (2.1). Tailsitters are common in both industry and research applications. The disadvantages are that the transition manoeuvre is highly complex, and during hover they are sensitive to cross winds.

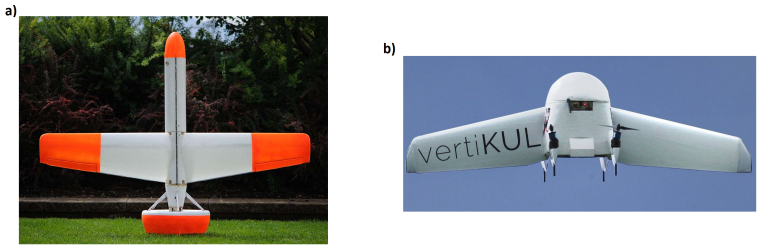

Figure 2.1: a) The Vertical Bat tail-sitter UAV [3]. b) VertiKUL tailsitter UAV [19]

2.1.0.2 Convertiplanes

Convertiplanes can be classified into several different sub-types, the three main being; tilt-rotor, tilt-wing and dual-system. The similarity between the types is that they all maintain the "normal" attitude of the aircraft body through the whole flight envelope, but they differ greatly in the mechanisms for transition from hover to cruise flight.

Tilt-rotor: Tilt-rotor aircraft start with some or all of the propellers facing upward for takeoff. After achieving vertical take-off the propellers are rotated to gain forward thrust until the propellers are aligned with the aircraft body and cruise speed is obtained. Even though the concept is common, and there exist a lot of research regarding these types of aircraft there exist so incredibly many different configurations and designs which make it difficult to get an overview. The designs vary in how many propellers, how many can rotate, are any of the propellers inserted as part of the ailerons or fuselage etc. Probably the first tilt-rotor UAV was Bell Eagle Eye which was inspired by the Bell Boeing V-22 Osprey which featured two rotating propellers at the edges of the wings of a more conventional aeroplane design [9, 10]. Another type is the TURAC UAV which is a flying wing with four propellers. Two coaxial propellers inserted as part of the fuselage assist with vertical thrust during hover, and two tilting propellers in front that will rotate during transition and provide thrust during cruising [31]. The Eagle Eye and the TURAC can be seen in Figure (2.2). The main disadvantages with the tilt-rotor design is the design complexity. Tilting of the rotors add additional actuators that complicate construction and assembly, and make it more difficult to perform larger design alterations.

Tilt-Wing: Til-wings are in many ways similar to tilt-rotors, but here the whole wing rotate along with the propellers when transitioning to and from

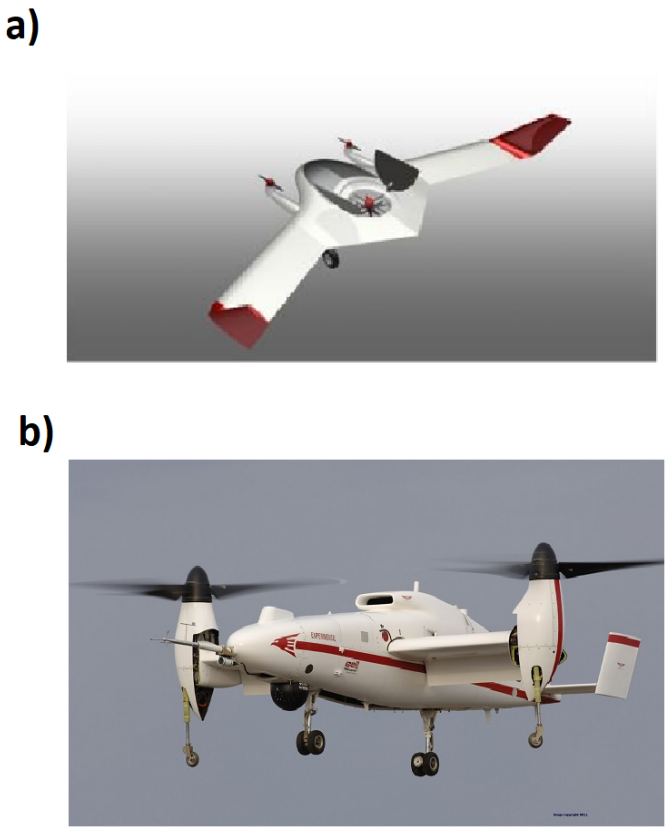

Figure 2.2: a) TURAC tilt rotor UAV [31]. b) Bell Eagle Eye [10]

cruise. In general what makes tilt-wings different from tilt-rotors is greater complexity in designs and larger sensitivity to cross-winds due to the tilted wings. Which results in some quite marvellous designs the such as NASA GL-10 Greased Lightning or the SAUVI [14, 29]. They can be seen in Figure (2.3)

Dual-Systems: The name *dual-systems* originates from the fact that these aircraft utilise two different propulsion systems. One for generating upward thrust and stabilise during take-off and landing, and one for generating forward speed during transition to cruise and during cruise flight. It is pretty much the expected result if a quadrotor was melted into a fixed-wing UAV. The main features of the dual-systems are simplicity in design and conceptually simple transition mechanism. The propulsion and control systems are however overly redundant, where fixed-wing systems will be completely passive during hover and vice versa for the quadcopter during cruising. As mentioned earlier, there exist little to no systematic research done on dual-propulsion convertiplanes despite several examples of commercial successes and many projects done by hobbyists with open source autopilots.

a)

b)

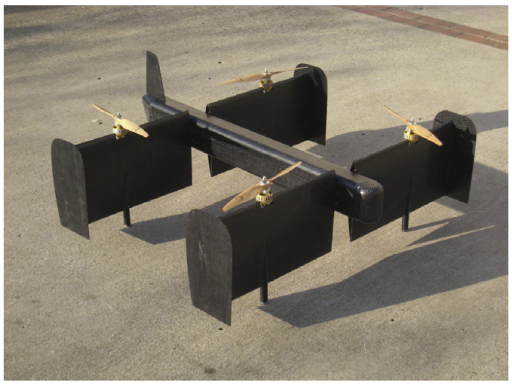

Figure 2.3: a) NASA GL-10 Greased Lightning [29]. b) SUAVI UAV [14].

2.2 Kestrel Design

There is a lot of research going towards *structronics*, which is a concept where the carrying structure of the UAV is used as part of the electronics and sensory system, instead of having the sensory mounted as an external payload. Kestrel as a concept is planned to be used at FFI as a testbed and sensor platform both in structronics research, and other fields of research where it might be useful to have an airborne sensor. Possible fields of interest are surveillance, reconnaissance, assistance during natural disasters, communication relays and probably a lot more. With structronics the idea will be to create a sensor, and then build an aircraft around it. This means that from design to design the aircraft might change a lot in order to accommodate the new sensor structure.

The initial objectives when Kestrel was designed was to build a fixed-wing vertical take-off and landing UAV for detection of electronic signals. It was also desired from FFI's side that the components were to be made of 3D printed nylon whenever possible in order to make it possible for it to house a sensory carrying structure in the future. In addition a set of requirements for

functionality and design were defined

- Endurance of approximately 3 hours
- Modular - For ease of maintenance and replacement
- Robustness - Be able to handle rough handling, transport and environmental conditions
- Portability - Transportable into the field with ease, and simple assembly.
- Low cost - To be able to have several UAVs simultaneously and cheap to repair.
- Payload - Must be able to accommodate a payload
- VTOL - Versatility
- Automation

With all the specifications above taken into account and weighed against each other it was the dual-system convertiplane concept that scored the highest [16]. This is the conceptually simplest design, and the one deemed to be most effective to accomplish the task. The design of Kestrel which can be seen in Figure (2.4) was heavily influenced by the Latitude HQ [22].

Figure 2.4: Kestrel Mk II during hover.

The fuselage is relatively small, only 450 *mm* long and 200 *mm* wide. It is 3D printed and designed to only hold the motor of the forward propeller with petrol tank, electronics including batteries for the four quadcopter propellers and the payload.

The main lifting surface during fixed-wing flight is the wings which are mounted onto the top of the fuselage. The aerofoil used is the National Advisory

Committee for Aeronautics (NACA) 2412 which has been successfully employed for many low speed aircraft [16]. The wing span is 2.5 m and winglets are mounted at the tips.

From the wings two parallel booms extend backwards towards the tail. The tail is the main stabilising surface of an aircraft. The structure of the tail is an A-tail design, and the aerofoil shape is NACA 0012. The A-tail shape means that the elevator and rudder will not be decoupled, but the ruddervator of the A-tail will have to perform both tasks simultaneously.

The fixed-wing forward level flight propeller is put in front of the vehicle and driven by a petrol engine. Petroleum has a much higher energy density than batteries which is needed in order to meet the requirement of prolonged flights.

The quadcopter propellers are mounted on parallel brooms that go from the wings back to the tail. They are situated in an X-formation around the fuselage of the aircraft. These, on the other hand, are electric meaning that there is an additional need for batteries to power them. This is quite a big UAV, lifting it solely by quadcopter thrust requires a lot of energy. With the weight inefficiency of batteries, and having two separate propulsion systems means there is a limitation to how long the UAV can maintain hovering. The initial Kestrel was designed to be able to maintain hovering for 5 minutes.

Chapter 3

Theory

In this chapter the theoretical basis used in this thesis are established. The first section defines the rules for notation, and the state variables that will be used, and their relation. Next the basics on rotation matrices and how to transform between various coordinate frames is presented, before going through the specific coordinate frames that will be used.

In the next section a detailed walk-through for modelling the Kestrel will be presented. First the kinematics and general rigid body dynamics will be derived before going through the most important external forces and moments acting on the UAV.

Lastly the basic theory for estimation and control is outlined before a detailed description of state feedback control as well as the MMA and LPV algorithms.

3.1 Mathematical Foundation

3.1.1 Notation

Scattered about in various literature are different conventions for notation. How mathematically accurate a text presented is needed to be depends a lot on what it is trying to do, and the usual conventions within that field of study. When the complexity of the work increases a more rigorously defined notation is required. The avoid confusion a quite detailed form of notation will be outline here.

3.1.1.1 Subscript and Superscript Nomenclature

- Coordinate frames will be denoted $\mathcal{F}^i$ or $\{i\}$ to represent coordinate frame i.

- Vectors are represented as lowercase bold font letters: $\boldsymbol{v}$. When represented in a specific coordinate frame it will be denoted by a superscript of the frame: $\boldsymbol{v}^i$.
- Coordinate axes unit vectors will be denoted by $\boldsymbol{i}^i, \boldsymbol{j}^i, \boldsymbol{k}^i$, with the same meaning of superscript.
- For angular velocities there will be a subscript denoting the rotating frame relative to a frame denoted in the first superscript. A second superscript will be used to indicate the frame from which this is represented. Example: frame $\mathcal{F}^a$ rotating relative frame $\mathcal{F}^b$ with angular velocity ω represented in frame $\mathcal{F}^c$ is denoted $\boldsymbol{\omega}_a^{bc}$.
- Rotation matrices from coordinate frame a to b will be denoted R_a^b, when rotating a specific angle there will also be parentheses indicating the angle $R_a^b(\eta)$.
- A hat over a vector will in most cases denote an estimate of the variable $\hat{\boldsymbol{x}}$
- Subscripts will be used in some occasions to specify either naming of a vectors such as the airspeed which will be denoted $\boldsymbol{V}_a$, or when specifying forces going along specific coordinate axes F_x^b where the subscript will indicate which Carthesian coordinate axis the force is working along.
- Scalars are denoted with lower-case non-bold letters such as a.

3.1.1.2 Euler Angles and Rotation Matrices

The Euler angles is one of several different ways of describing the orientation of a frame relative to another. The orientation of one Carthesian coordinate with respect to another can always be described by three successive rotations around the orthogonal coordinate axes, and these rotation angles are called the Euler angles. The rotation matrices describing the rotation around each of the coordinate axes are given by

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}$$
$$R_y(\theta) = \begin{bmatrix} c_\phi & 0 & s_\phi \\ 0 & 1 & 0 \\ -s_\phi & 0 & c_\phi \end{bmatrix}$$
$$R_y(\psi) = \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where the subscripts x, y, z denote the coordinate axis the rotation was done around, and $c_{(\cdot)} = \cos(\cdot)$ and $s_{(\cdot)} = \sin(\cdot)$. There are various conventions of

which order to make the rotations depending on the specific field of science. Probably the most used, and the ones used withing aerospace engineering are the ZYX and XYZ Euler angles[1]. Those are the ones that will be used in this thesis. This results in the rotation matrix from one coordinate frame $\{a\}$ to another $\{b\}$ is given by

$$R_a^b = R_z(\psi)R_y(\theta)R_x(\phi) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\phi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{3.1}$$

3.1.2 Coordinate Frames

In order to describe the motion of a moving body two things are required; forces and moments acting on the body that cause movement, and a stated reference frame from which the movement is viewed from. The various forces are often more easily defined in different coordinate frames, and thus it is necessary to have a complete understanding of the various coordinate frames, and how to transform between them in order to represent the motions.

3.1.2.1 Inertial frame

An inertial frame is easily described by; a coordinate frame where Newton's laws of motion can be described in their most simple form. Strictly speaking, a coordinate frame defined on earth's surface will not be an inertial frame. When operating within reasonably small ranges, assuming the earth as flat in a local area, serves in most operations as a good approximation for an inertial frame. Which is called the *local tangent plane*-approximation.

However, when flying long distances at a constant altitude, it will no longer be sufficient to model the earth as flat. Due to earths curvature the trajectory of the plane will have to be curved along with earths surface, and due to earths rotation the plane's movement relative to the earth surface would seem to rotate due to the Coriolis effect. In these cases it would be necessary to use an earth centred coordinate frame and take into account rotation and curvature of the earth.

In this work it will be assumed that the UAV will travel short enough distances that the local tangent plane located around the UAV serves the purpose as an inertial frame. This frame is denoted $\{i\}$. It is aligned such that the $\boldsymbol{i}^i$-axis points to the north, $\boldsymbol{j}^i$ to the east and the $\boldsymbol{k}^i$-axis points downwards towards the earth's centre, this is also known as a North-East-Down(NED) frame.

[1]Fun fact: XYZ and ZYX rotation orders are actually part of what is called *Tait-Bryan*-angles. This naming is rarely used, and in this thesis the Euler-angles name will be used to avoid confusion.

3.1.2.2 Vehicle Frame

The vehicle frame v is centred in the *centre of mass*(CM) of the UAV, and the axes are aligned with the earth frame. It is most often used for defining navigation equations relative to the earth-frame origin. The inertial frame and vehicle frame is illustrated in Figure (3.1)

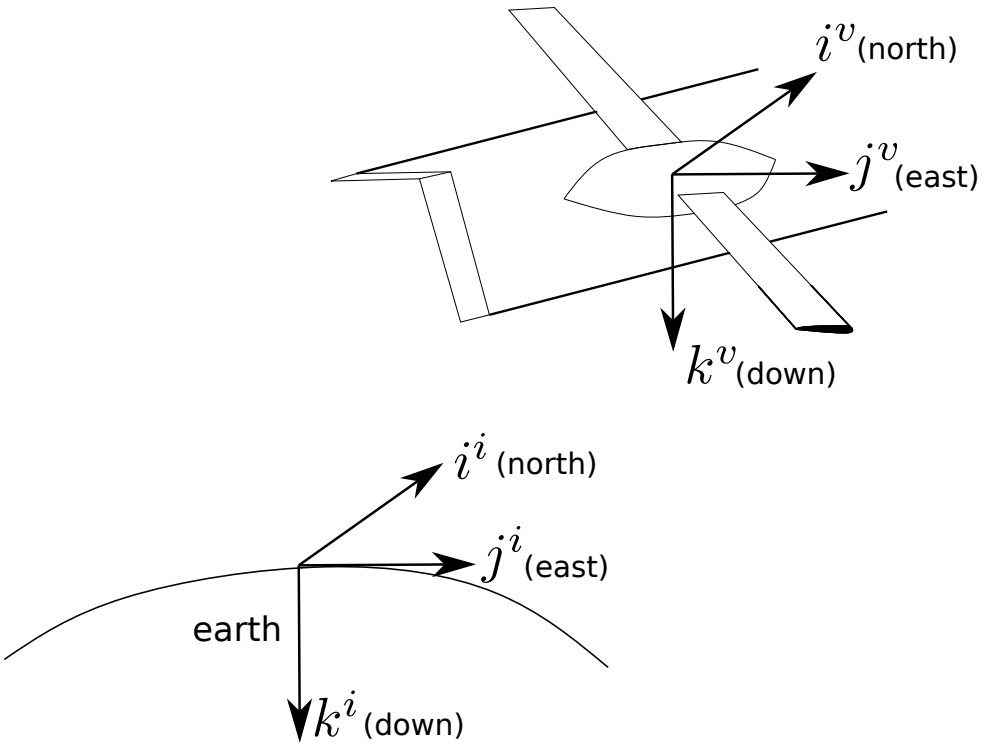

Figure 3.1: The relationship between the local tangent plane inertial frame and the vehicle frame centred at the CM of the UAV.

3.1.2.3 Vehicle-1 Frame

The vehicle-1($\{v_1\}$) frame is the same as the vehicle frame, except that it is rotated an angle ψ around the $\boldsymbol{k}^v$-axis such that the $\boldsymbol{i}^1$-axis points in the direction on a compass the UAV is heading. The transformation from $\{v\}$ to $\{v_1\}$ is given by

$$\boldsymbol{r}^{v_1} = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{r}^v = R_v^{v_1} \boldsymbol{r}^v \tag{3.2}$$

3.1.2.4 Vehicle-2 Frame

Again, the vehicle-2($\{v_2\}$) frame is the same as the $\{v_1\}$-frame except that it is rotated a pitch angle θ around the $\boldsymbol{j}^{v_1}$-axis. The transformation is given by

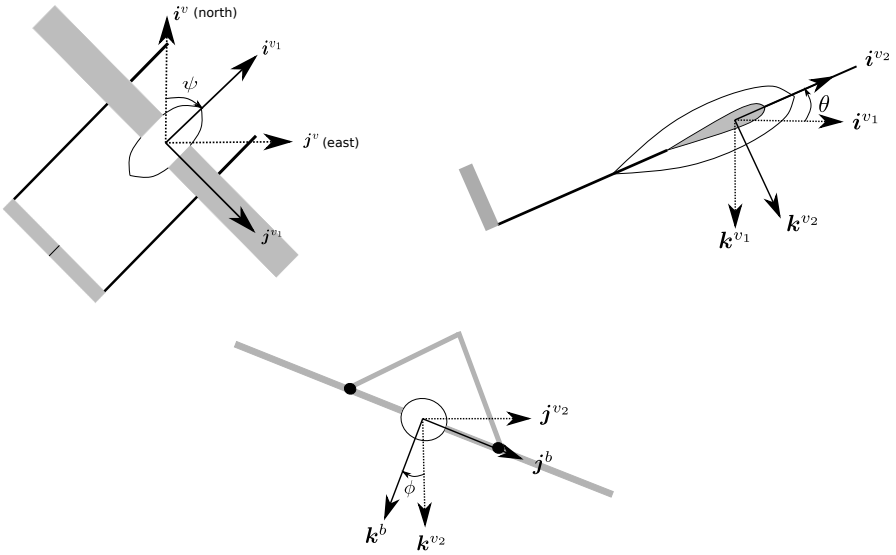

Figure 3.2: The step-wise rotations over the Euler-angles from the vehicle frame to the body frame

$$\boldsymbol{r}^{v_2} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \boldsymbol{r}^{v_1} = R_{v_1}^{v_2} \boldsymbol{r}^{v_1} \tag{3.3}$$

3.1.2.5 Body Frame

The body frame is the frame which is centred at the CM of the UAV and aligned with the orientation of the UAV. It is obtained by rotating a roll angle ϕ around the $\boldsymbol{i}^{v_2}$ axis. The transformation from the $\{v_2\}$-frame is given by

$$\boldsymbol{r}^{v_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \boldsymbol{r}^{v} = R_{v}^{v_1} \boldsymbol{r}^{v} \tag{3.4}$$

This is the frame most of the forces and motions will be defined from, and now a nice transformation from the vehicle from to the body frame is available by utilising the transformations presented above.

$$R_v^b(\phi,\theta,\psi) = R_{v_2}^b R_{v_2}^{v_1} R_v^{v_1} = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_{psi} + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_{psi} + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_{psi} & c\phi c_\theta \end{bmatrix} \tag{3.5}$$

Which is the transpose of the rotation matrix described in Equation(3.1). Trying to solve the Euler angles rotation matrix for the respective roll, pitch and yaw angles will reveal that there will be singularities for $\theta \pm 90$ degrees, this is called gimbal lock. As long as no aggressive manoeuvres or acrobatics is attempted this will not be an issue. If needed, gimbal lock problems can be solved by using quaternions instead.

3.1.2.6 Stability Frame

The aerodynamic effects are the results of the aircraft's movement relative to the surrounding air. This relative velocity between the aircraft and the air is named the airspeed vector $\boldsymbol{V}_a$. Rotating the body frame an angle α about the $\boldsymbol{j}^b$-axis give stability frame. α is called the angle of attack, and describes the pitch at which the aircraft enters the wind. It is one of the more important parameters for generating lift, and for most aerofoils it must be positive in for the aircraft to generate lift upwards. The rotation matrix is given by

$$r^s = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} r^b = R_b^s r^b \tag{3.6}$$

3.1.2.7 Wind Frame

Finally, to obtain the wind frame which is aligned with the wind-vector the stability-frame can be rotated an angle β around the $\boldsymbol{k}^s$-axis. The β angles is more commonly referred to as the sideslip angle, and is important when analysing the longitudinal and lateral aerodynamic forces and moments. The rotation matrix is given by

$$\boldsymbol{r}^w = \begin{bmatrix} \cos(\beta) & \sin(\beta) & 0 \\ -\sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{r}^s = R_s^w \boldsymbol{r}^s \tag{3.7}$$

Putting the transformations together and inverting it gives the transformation from the wind frame to the body frame

$$R_w^b(\alpha,\beta) = (R_s^w R_b^s)^T = \begin{bmatrix} c_\beta c_\alpha & -s_\beta c_\alpha & -s_\alpha \\ s_\beta & c_\beta & 0 \\ c_\beta s_\alpha & -s_\beta s_\alpha & c_\alpha \end{bmatrix} \tag{3.8}$$

Name	Description
p_n	North position of UAV in inertial frame $\mathcal{F}^i$ along$\boldsymbol{i}^i$
p_e	East position of UAV in inertial frame $\mathcal{F}^i$ along$\boldsymbol{j}^i$
p_d	Down position of UAV in inertial frame $\mathcal{F}^i$ along$\boldsymbol{k}^i$
u	Linear velocity of UAV measured along $\boldsymbol{i}^b$ in body frame $\mathcal{F}^b$
v	Linear velocity of UAV measured along$\boldsymbol{j}^b$ in body frame $\mathcal{F}^b$
w	Linear velocity of UAV measured along$\boldsymbol{k}^b$ in body frame $\mathcal{F}^b$
ϕ	Roll angle around $\boldsymbol{i}^{v2}$ in vehicle-2 frame $\mathcal{F}^{v2}$
θ	Pitch angle around $\boldsymbol{j}^{v1}$ in vehicle-2 frame $\mathcal{F}^{v1}$
ψ	Yaw angle around $\boldsymbol{k}^v$ in vehicle-2 frame $\mathcal{F}^v$
p	Roll rate measuring around $\boldsymbol{i}^b$ in body $\mathcal{F}^b$
q	Pitch rate measuring around $\boldsymbol{j}^b$ in body $\mathcal{F}^b$
r	Yaw rate measuring around $\boldsymbol{k}^b$ in body $\mathcal{F}^b$

Table 3.1: The state variables of the UAV defined in the respective frames they are used.

3.1.3 State Variables

An UAV can move in any direction in 3D space. When deriving the equations of motion twelve equations will be required, and thus twelve state variables. A list of these can be seen in Table (3.1)

The position of the UAV is defined in the inertial frame $\mathcal{F}^i$ and is denoted $\boldsymbol{p}^i = [p_n, p_e, p_d]^T$. Strictly speaking the variable should be denoted with an $\{i\}$ superscript, but it is omitted to have cleaner looking equations, if the letters are represented in any other frame or context proper notation will be utilised. Defining the altitude with *down* direction is not very intuitive since the altitude would be negative for a position above ground. For that reason altitude is often given by $h^i = -p_d$ in plots etc.

The linear velocities $\boldsymbol{v}^b = [u, v, w]^T$ are given in the body frame. This is because most of the forces affecting UAVs are more easily described in the body frame. The transformation is instead done on the body-defined velocities to inertial position rates, which will be seen when deriving the linear kinematics. The naming convention used for the linear velocities is what is used in most textbooks and superscripts are omitted to avoid unnecessary clutter, proper notation will be used if needed to avoid confusion in certain settings.

The attitude of the UAV is described by the roll, pitch and yaw angles $\boldsymbol{\Theta} = [\phi, \theta, \psi]^T$. In reality these are presented in the intermediate frames during the rotation from body-frame to the inertial frame. They are not given any superscript since the variables will not be used for anything else than describing the Euler angles, and a superscript would probably only create unnecessary clutter.

The angular rates $\boldsymbol{\omega}_b^{vb} = [p, q, r]$ are also described in the body-frame for the

same reason as the linear rates; the moment descriptions become easier. Same argument for omitting superscripts apply here.

3.2 Kinematics and Dynamics

In order to be able to develop good control strategies for the Kestrel that can be verified in simulation, an accurate dynamic model must be in place.What is meant by an accurate dynamic model is that it contains the most relevant forces and moments in the flight regime under analysis. Especially understanding the aerodynamic effects, and the ability to model these are important when designing and assessing the controller to ensure that the reality gap when transitioning from simulation to the actual UAV is as small as possible.

3.2.1 Assumptions

Below is a list of assumptions that will be taken into account when deriving kinematic and dynamic models for the Kestrel UAV.

1. The UAV is a rigid body
2. The mass will remain constant
3. The UAV is symmetric about the $\boldsymbol{i}^b\boldsymbol{k}^b$-plane.
4. The quadcopter propellers have a fixed pitch relative to the body frame, and the thrust vector from each propeller points parallel with the $\boldsymbol{k}^b$-axis and in the opposite direction while hovering.
5. The forward propeller thrust vector is parallel with the $\boldsymbol{i}^b$-axis and in the same direction.
6. Propellers with odd indices rotate counterclockwise and even index propellers rotate clockwise.

3.2.2 Kinematics

Rigid-body kinematic modelling is the method of studying the movement of the body independent of the external forces involved. Instead it takes in velocities and relates them to positions. When regarding the translational motion the position of the UAV is usually represented in the inertial frame by $\boldsymbol{p}^i$, while the translational velocity $\boldsymbol{v}^b$ usually is represented in the body frame. These two are related by a differentiation and a rotation matrix

$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{bmatrix} = (R_v^b)^T \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.9}$$

where $(R_v^b)^T$ is given by Equation (3.5).

The same problem arises when considering rotational motions, where angular rates $\omega_b^{vb} = \left[p, q, r\right]^T$ are usually represented in the body system. Following the derivations in Sections(3.1.2), the angular positions, or Euler angles are all represented in the different coordinate frames. Using the transformations between the various coordinate frames the body frame angular rates can be represented by

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_{v2}^b(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{v2}^b R_{v1}^{v2}(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}. \tag{3.10}$$

Adding these terms together and inverting the total transformation matrix, yields a relationship that expresses the time derivative of angular positions $\boldsymbol{\Theta}$ in terms of angular positions and body rates

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.11}$$

3.2.3 Rigid-Body Dynamics

In the subject of dynamics the goal is to explain how external forces acting upon the body causes movement. Combining the equations derived earlier a set of equations describing the full six degrees of freedom motion of rigid aerospace vehicle will be arrived at. Looking up any source on rigid body dynamics will show that any movement can be decomposed into rotational and translational dynamics that can be treated separately. In these sections, first the Euler equations for rotational motions are derived, and then the translational equations of motion are derived. Later the most important forces and moments acting upon the body will be explained.

3.2.4 Rotational Dynamics

Denote the angular momentum of a rigid body about its CM taken in an inertial frame as $\boldsymbol{h}$. Looking up any textbook on classical mechanics will give that the time derivative of the angular momentum taken in the inertial frame is equal to the sum of all vector moments $\boldsymbol{M}^i = [L^i, M^i, N^i]$ acting on the body.

$$\boldsymbol{M}^i = \dot{\boldsymbol{h}}_{CM}^{ii} \tag{3.12}$$

As described earlier, most moments affecting the UAV are more easily described in the body frame. A mass element δm at position $\boldsymbol{r}^b$ will have an angular momentum about CM in the body frame

$$\delta \boldsymbol{h}_{CM} = \boldsymbol{r}^b \times (\boldsymbol{v}^i_{CM} + \boldsymbol{\omega}^{ib}_b \times \boldsymbol{r}^b)\, \delta m \tag{3.13}$$

Choosing to use the body frame for integrating up Equation (3.13) it can be shown that the resulting angular momentum vector becomes

$$\boldsymbol{h}^b_{CM} = \iiint \begin{bmatrix} (y^2+z^2) & -xy & -xz \\ -xy & (x^2+z^2) & -yz \\ -xz & -yz & (x^2+y^2) \end{bmatrix} dm \boldsymbol{\omega}^{ib}_b \tag{3.14}$$

$$= \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & -J_{yz} & J_{zz} \end{bmatrix} \boldsymbol{\omega}^{ib}_b = J^b \boldsymbol{\omega}^{ib}_b \tag{3.15}$$

In reality J^b is dependent on the reference frame, but no other case than the inertia being calculated in the body-frame will be considered, thus the superscript will be omitted. Under the assumptions stated earlier in this section the inertia matrix J is constant. This however is a truth with modifications as the mass of the *Kestrel* will change as fuel is burned, and any bending of the wings during flight will effectively change the mass distribution and the inertia matrix. For a typical UAV, and what is also the case for the Kestrel is that the UAV is symmetric about the $\boldsymbol{i}^b\boldsymbol{k}^b$-plane. This will result in all off-diagonal elements in the inertia matrix that are integrals over the $\boldsymbol{j}^b$-axis will become zero.

Inserting Equation (3.14) into Equation (3.12) and noting the difference in reference frames yields

$$\boldsymbol{M}^b = \dot{\boldsymbol{h}}^{ii}_b = \dot{\boldsymbol{h}}^{ib}_b + \boldsymbol{\omega}^i_b \times \dot{\boldsymbol{h}}^{ib}_b \tag{3.16}$$

$$= J_b \dot{\boldsymbol{\omega}}^{ib}_b + S(\boldsymbol{\omega}^{ib}_b) J_b \boldsymbol{\omega}^{ib}_b \tag{3.17}$$

Where $S(\boldsymbol{\omega}^{ib}_b)$ is the skew-symmetric matrix denoting $(\boldsymbol{\omega}^{ib}_b \times)$. This can be written out to give the complete representation of the rotational dynamics of the UAV

$$L^b = J_{xx}\dot{p} + (J_{zz} - J_{yy}) + J_{xz}(\dot{r} + pq) \tag{3.18}$$

$$M^b = J_{yy}\dot{q} + (J_{xx} - J_{zz}) + J_{xz}(r^2 - p^2) \tag{3.19}$$

$$N^b = J_{zz}\dot{r} + (J_{xx} - J_{yy}) + J_{xz}(\dot{p} + qr) \tag{3.20}$$

Solving this for the body angular rates $[\dot{p}, \dot{q}, \dot{r}]$ gives

$$\begin{aligned}
\dot{p} =& \frac{1}{\Gamma} J_{xz}[J_{xx} - J_{yy} + J_{zz}]pq - [J_{zz}(J_{zz} - J_{yy}) + J_{xz}^2]qr \\
&+ J_{zz}L^b + J_{xz}N^b \\
\dot{q} =& \frac{1}{J_{yy}}([J_{zz} - J_{xx}]rp - J_{xz}[p^2 - r^2] + M^b) \\
\dot{r} =& \frac{1}{\Gamma}(-J_{xz}[J_{xx} - J_{yy} + J_{zz}]qr + [J_{xx}(J_{xx} - J_{yy}) + J_{xz}^2]pq \\
&+ J_{xx}N^b + J_{xz}L^b)
\end{aligned} \tag{3.21}$$

where $\Gamma = J_{xx}J_{zz} - J_{xz}^2$. A typical quadrotor helicopter will also have a plane of symmetry about the $y_b z_b$-plane thus resulting in $J_{xz} = 0$ and the inertia matrix becomes diagonal. This make it possible to simplify Equation (3.21) even further

$$\begin{aligned}
\dot{p} =& \frac{J_{yy} - J_{zz}}{J_{xx}}qr + \frac{L^b}{J_{xx}} \\
\dot{q} =& \frac{J_{zz} - J_{xx}}{J_{yy}}qr + \frac{M^b}{J_{yy}} \\
\dot{r} =& \frac{J_{xx} - J_{yy}}{J_{zz}}qr + \frac{N^b}{J_{zz}}
\end{aligned} \tag{3.22}$$

3.2.5 Translational Dynamics

In the same manner as with the rotational dynamics Newton's second law states, with a small addition, that the time derivative of the linear momentum in an inertial frame of a rigid body will equal the sum of external forces acting upon the body.

$$\boldsymbol{F}^i = m\dot{\boldsymbol{v}}_{CM}^i \tag{3.23}$$

Depending on what desired to achieve, this equation can be transformed into various coordinate systems to make mathematics or representations more understandable. In this thesis the body system is chosen, and all forces acting upon the body will have to be transformed to that reference frame. Using differentiation of vectors in different frames and the fact that differentiating angular rates is independent of the choice of reference frame gives the dynamic equations for a rigid body moving in an inertial frame, represented in the body frame

$$\boldsymbol{F}^b = m\left(\dot{\boldsymbol{v}}_b^b + \boldsymbol{\omega}_b^{ib} \times \boldsymbol{v}_b^b\right) \tag{3.24}$$

Expanding the cross product yields

$$\begin{aligned} F_x^b &= m(\dot{u} + qw - rv) \\ F_y^b &= m(\dot{v} + ru - pw) \\ F_z^b &= m(\dot{w} + pv - qu) \end{aligned} \tag{3.25}$$

This can be rewritten into a representation more suitable for mathematical modeling

$$\begin{aligned} \dot{u} &= (rv - qv) + \frac{F_x^b}{m} \\ \dot{v} &= (pw - ru) + \frac{F_y^b}{m} \\ \dot{w} &= (qu - pv) + \frac{F_z^b}{m} \end{aligned} \tag{3.26}$$

3.2.6 Forces and Moments

The full 6 degrees of freedom equations of motion above only describe the motions as they would be affected by external forces and moments without specifying any details regarding the actual forces that interact with the UAV. In this section the largest forces and moments acting upon Kestrel will be investigated and how they vary with the state of the system. There are quite big differences in what to consider when modelling a fixed-wing aircraft versus a quadrotor. The largest difference in the consideration of aerodynamic forces; quadrotor models usually neglect it, while the functionality of fixed-wing aircraft are founded on the very concept. There will also be considerations regarding effects that may arise during the transition phase.

3.2.6.1 Gravitational Force

Assuming that the altitude the UAV operates in is within reasonable altitude the magnitude of gravitational pull can be assumed constant. Represented in the body fixed frame it is given by

$$F_g^b = R_i^b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = mg \begin{bmatrix} -\sin\theta \\ \sin\phi\cos\theta \\ \cos\theta\cos\phi \end{bmatrix} \tag{3.27}$$

3.2.6.2 Propulsion Forces

With this dual propulsion system that is in the Kestrel UAV it is important to have a good understanding of how the forces and moments generated through the various propellers affect the dynamics of the UAV. To get a complete understanding of how the rotation of the propellers generate forces and moments is nigh impossible

Front Propeller Force and Momentum In the Kestrel and in general most fixed-wing UAV the propulsion system will generate a collective force that will go through the UAV CM and along the x_b-axis. There will not be a detailed examination of the deeper mechanics of the propeller here. One approach is to say that the total force from the propeller is the pressure difference of the air upstream of the propeller and downstream of the propeller multiplied by the area of the propeller. Using Bernoulli's principle the pressures will depend on the airspeed before and after the propeller. The upstream air velocity is the airspeed, and the downstream velocity will depend on the angular velocity of the propeller. Which gives

$$V_{exit} = k_{motor}\delta_t, \tag{3.28}$$

if it is assumed that there is a linear relationship between the throttle command δ_t and the angular velocity through a motor constant k_{motor}. Putting all of this together yields an expression of the propeller linear force

$$\boldsymbol{F}_p = \frac{1}{2}\rho S_{prop} C_{prop} \begin{bmatrix} (k_{motor}\delta_t)^2 - V_a^2 \\ 0 \\ 0 \end{bmatrix} \tag{3.29}$$

where ρ is the air density, C_{prop} is a rotor specific aerodynamic constant and S_{prop} is the propeller area. It can also be assumed that the angular velocity of the propeller is so high that the effects of spiralling slipstream can be neglected. The effects of gyroscopic moments arising from angular movement of the rotating propeller will not be considered. It may however become an important effect especially with larger propellers [34, 35].

Quadcopter Propellers The same assumptions that were made regarding the forward facing propeller can be done with the quadcopter propellers. But instead of regarding the resulting force and reaction torque as a function of the thrust command, they will be expressed as functions of the propeller constants C_{Tq} and C_{Mq} for force and moment respectively and the angular speed ω_i

$$\begin{aligned} \boldsymbol{F}_i &= C_{Tq}\omega_i^2 \\ \boldsymbol{M}_i &= C_{Mq}\omega_i^2. \end{aligned} \tag{3.30}$$

Later when discussing the controllers they will output desired roll, pitch and yaw commands, and the angular velocities of each propeller can be resolved from these. There has been made a distinction in the equations between the two to not cause confusion and also emphasise the importance of the actual angular velocities and their directions on the various moment commands.

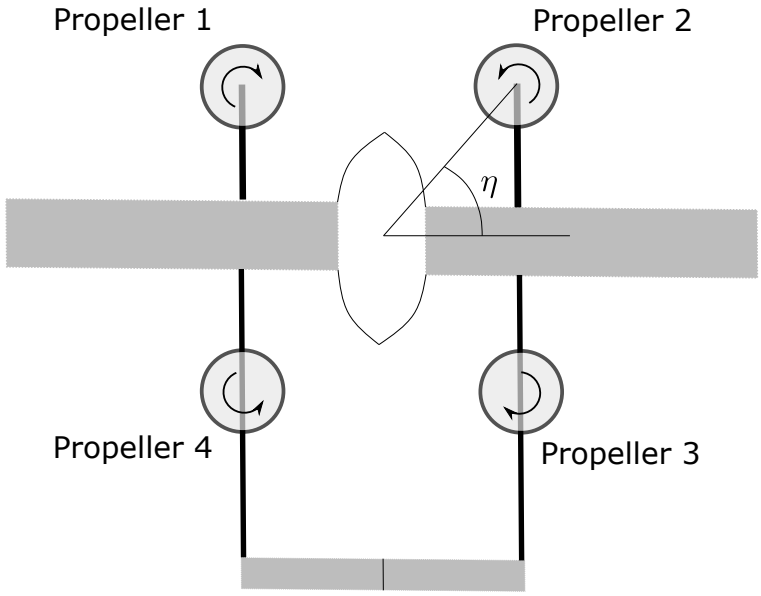

Figure 3.3: The geometry and rotation directions of the quadcopter propellers.

Looking at Figure (3.3) gives an understanding of the configuration of the quadcopter propellers and their rotation directions. It can be seen that propellers 1 and 3 rotates counterclockwise while 2 and 4 rotate clockwise, and they are all a distance d from the CM and the angle between the line out to each propeller and the $\boldsymbol{i}^b$-axis is η.

Still using the rigid body assumptions means that any bending of the rods going to the quadcopter propellers will not be considered. This means that it is assumed that all thrust forces of the quadcopter propellers will be parallel to the $\boldsymbol{k}_b$-axis. The total thrust produced by the propellers are given by the sum of all the force generated by the propellers. The roll and pitching moments are given by the difference between forces applied by the propellers across the respective axes. Lastly the yaw moment is given by the difference in reaction torques generated by the opposite rotating propellers

$$\begin{bmatrix} F_{Tq}^b \\ \tau_{roll} \\ \tau_{pitch} \\ \tau_{yaw} \end{bmatrix} = \begin{bmatrix} C_{Tq}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ C_{Tq}\cos(\eta)(\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) \\ C_{Tq}\sin(\eta)(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \\ C_{Mq}(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2). \end{bmatrix} \tag{3.31}$$

3.2.6.3 Aerodynamic Forces and Moments

Aerodynamics is at its core quite simple. An aerofoil goes through the air, and due to asymmetries of the airflow above and below the aerofoil there will be varying pressures on the aerofoil surface. When summed up, these pressure differences give a resulting force acting upon the aerofoil. The simplicity however goes downhill from there. There is almost no limit to how detailed descriptions of the aerodynamics can become. In this thesis things will be kept sufficiently detailed to get a good understanding of relevant forces and moments, but no more than that. It is common to decompose the aerodynamic forces into lateral and longitudinal forces and moments.

Longitudinal Aerodynamic Forces: Longitudinal aerodynamic forces are probably the first people think of when mentioning aerodynamics. These are the lift, drag and pitching moments that cause movement in the $\boldsymbol{i}^b\boldsymbol{k}^b$-plane. Holding up a flat surface up against the wind it can be confirmed that the most important elements to the forces is the air speed across the surface, the angle of attack against the wind and the surface area. In addition these forces will also depend on the density of the air ρ, the pitch rate q and the elevator deflection δ_e.

$$\begin{aligned} F_{lift} &= \frac{1}{2}\rho V_a^2 S C_L(\alpha, q, \delta_e) \\ F_{drag} &= \frac{1}{2}\rho V_a^2 S C_D(\alpha, q, \delta_e) \\ M &= \frac{1}{2}\rho V_a^2 S c C_M(\alpha, q, \delta_e) \end{aligned} \tag{3.32}$$

where C_L,C_D,C_M are non-dimensional aerodynamic coefficients that depend upon said parameters, S is the aerodynamic surface area, and c is the mean chord of the aerofoil.

In general, these equations are nonlinear. However, as long as the flow around the surface remains laminar(i.e. small angles of attack) these forces and moments can up to a certain degree be modelled using linear approximations. A first order Taylor expansion of the equations can be written as

$$
\begin{aligned}
F^s_{lift} =& \frac{1}{2}\rho V_a^2 S(C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e) \\
F^s_{drag} =& \frac{1}{2}\rho V_a^2 S(C_{D_0} + C_{D_\alpha}\alpha + C_{D_q}\frac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e) \\
M^b =& \frac{1}{2}\rho V_a^2 Sc(C_{M_0} + C_{M_\alpha}\alpha + C_{M_q}\frac{c}{2V_a}q + C_{M_{\delta_e}}\delta_e)
\end{aligned}
\tag{3.33}
$$

The parameters C_{L_α} and C_{L_q} are commonly referred to as stability derivatives while $C_{L_{\delta_e}}$ is called the control derivative. The naming derivative arise from the fact that the coefficients come from the partial derivatives of the Taylor expansion. C_{lift_0} is the value of C_{lift} when $q = \alpha = \delta_e = 0$. The lift and drag forces in Equation (3.33) are expressed in the stability frame. They are given in the body frame by a rotational transformation

$$
\begin{bmatrix} F^b_x \\ F^b_z \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} -F^s_{drag} \\ -F^s_{lift} \end{bmatrix}. \tag{3.34}
$$

It must again be emphasised that these linear approximations are only valid within a quite small angle of attack when the airflow over the wings is laminar. In this thesis it is will be assumed that these assumptions hold. Details regarding approaches to take when these assumptions do not hold can be found in more rigorous works on UAV aerodynamics [35, 46].

Lateral Aerodynamic Forces: The lateral aerodynamic forces are less intuitive ones for a layman. They are the ones that cause movement along the y_b-direction as well as cause rolling and yawing movements. As would expected the most important factor for these forces and moments is the sideslip angle β. Sideslip is the angle at which the wind hits the aircraft. But, in addition they will also depend on the deflections of the ailerons and ruddervator, as well as the roll and yaw rates. In the same manner as the longitudinal forces and moments a general expressions can be made for the lateral forces and moments.

$$
\begin{aligned}
F^b_y =& \frac{1}{2}\rho V_a S C_y(\beta, p, q, r, \delta_a, \delta_r) \\
L^b =& \frac{1}{2}\rho V_a S b C_l(\beta, p, q, r, \delta_a, \delta_r) \\
N^b =& \frac{1}{2}\rho V_a S b C_N(\beta, p, q, r, \delta_a, \delta_r).
\end{aligned}
\tag{3.35}
$$

Here C_y, C_l and C_N are non-dimensional aerodynamic coefficients. b is the wingspan. Also here it will be assumed that the UAV will not stray too far from nominal flight conditions such that a linear approximation will yield and acceptable accuracy. Thus with a first order Taylor expansion a linear expression can be achieved for the lateral forces and moments

$$
\begin{aligned}
F_y^b =& \frac{1}{2}\rho V_a^2 S(C_{y_0} + C_{y_\beta}\beta + C_{y_p}\frac{b}{2V_a}p + C_{y_r}\frac{b}{2V_a}r + C_{y_{\delta_a}}\delta_a + C_{y_{\delta_r}}\delta_r) \\
L^b =& \frac{1}{2}\rho V_a^2 Sb(C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r) \\
N^b =& \frac{1}{2}\rho V_a^2 Sb(C_{N_0} + C_{N_\beta}\beta + C_{N_p}\frac{b}{2V_a}p + C_{N_r}\frac{b}{2V_a}r + C_{N_{\delta_a}}\delta_a + C_{N_{\delta_r}}\delta_r).
\end{aligned} \tag{3.36}
$$

The parameters in these equations are again the results of partially differentiating the coefficients with respect to their respective parameters denoted in the subscript. It should be noted that the parameters C_{lat_0}, C_{L_0} and C_{N_0} will in general be zero for a symmetric aircraft.

Propeller Aerodynamics Most often the propellers of a quadcopter are assumed to be rigid bodies, and that the thrust generated by each propeller is only dependent on the voltage applied to the motor and thus the angular velocity of the blade. There are mainly two effects that occur due to motion of the UAV relative the surrounding air that may be especially relevant for an aircraft such as the Kestrel. These can severely effect the generated thrust from the propellers [20].

The first effect is due to how the effectively generated thrust from the propellers are affected by the airspeed V_a across the blades and the angle of attack α. The ideal thrust T generated by a power input P is given by

$$
T = \frac{P}{V_a \sin\alpha + V_i} \tag{3.37}
$$

where V_i is the change in airspeed induced by the rotor blades with respect to the airspeed V_a. It is clear when analysing the equation that during slow translational movements and low angles of attack the effect on the total thrust is small. During the transition phase of the Kestrel to ensure that the aircraft does not loose too much altitude the quadcopter propellers will have to assist with generating lift and stabilising the aircraft until the fixed-wing aerodynamics and control can take over. This will require the quadcopter propellers to operate at quite substantial translation velocities. During this phase the effective power required to maintain a constant thrust will, according to Equation(3.37), be heavily influenced by the forward speed and angle of attack.

The second effect is what is called *blade flapping*. The propellers are aerodynamic surfaces, and the total lift generated by the aerodynamic surface depends on its velocity relative to the surrounding air. During horizontal movement the advancing part of the propeller will see a larger velocity relative the surrounding air than the retracting part of the propeller. This means that the two sides will generate different lift, and cause a flapping of the rotors once per rotation. The maximal upward and downward displacements of the rotors will

happen at forward and backward position respectively relative the horizontal velocity. This effect will be equal on all propellers such that any induced torques on a single propeller will effectively be cancelled out by the other propellers. If the blade flapping cause an effective tilting of the propeller blades by an angle μ, the propellers will generate a longitudinal thrust $F_{Tq}^b \sin(\mu)$. Also the effective thrust in the $\boldsymbol{k}^b$-direction will be reduced to $F_{Tq}^b \cos(\mu)$ [20, 27, 34].

These aerodynamic effects on the quadcopter propellers will not be investigated in further detail here, but it is important to mention their existence. Usually when considering quadcopters they can be neglected due to the low horizontal velocities. During transitional manoeuvres of the Kestrel the effects may become prevalent, and if the quadcopter propellers used to generate lift during this phase, the effective force generated from the propellers might be severely impeded and may cause degradation of the stability of the aircraft.

3.3 Summary of the Dynamic Modelling of the Hybrid UAV

This section summarises the equations obtained from the dynamic modelling in the previous sections for modelling a dual system convertiplane. The resulting system is in essence a fixed-wing aircraft with the added capabilities from the quadcopter propellers.

3.3.1 Dynamic Model

Without considering the specifics of the external forces the six degrees of freedom 12-state dynamic model for a general rigid body can be expressed as

$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{bmatrix} = \begin{bmatrix} c_\theta c_{psi} & s_{phi} s_\theta c_\psi - c_\phi s_\psi & c_{phi} s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.38}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m} \begin{bmatrix} F_x^b \\ F_y^b \\ F_z^b \end{bmatrix} \tag{3.39}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.40}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Gamma}(J_{xz}(J_{xx} - J_{yy} + J_{zz})pq - [J_{zz}(J_{zz} - J_{yy}) + J_{xz}^2]qr \\ +J_{zz}L + J_{xz}N) \\ \frac{1}{J_{yy}}((J_{zz} - J_{xx})rp - J_{xz}(p^2 - r^2) + M) \\ \frac{1}{\Gamma}(-J_{xz}(J_{xx} - J_{yy} + J_{zz})qr + [J_{xx}(J_{xx} - J_{yy}) + J_{xz}^2]pq \\ +J_{xx}N + J_{xz}L) \end{bmatrix} \tag{3.41}$$

3.3.2 External Forces and Moments

The external forces can be summarised as:

$$\begin{aligned} \begin{bmatrix} F_x^b \\ F_y^b \\ F_z^b \end{bmatrix} &= \begin{bmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \end{bmatrix} \\ &+ \frac{1}{2}\rho V_a^2 S \begin{bmatrix} C_x(\alpha) + C_{x_q}\frac{c}{2V_a}q + C_{X_{\delta_e}}(\alpha)\delta_e \\ C_{y_0} + C_{y_\beta}\beta + C_{y_p}\frac{b}{2V_a}p + C_{y_r}\frac{b}{2V_a}r + C_{y_{\delta_a}}\delta_a + C_{y_{\delta_r}}\delta_r \\ C_z(\alpha) + C_{z_q}(\alpha)\frac{c}{2V_a}q + C_{z_{\delta_e}}(\alpha)\delta_e \end{bmatrix} \\ &+ \frac{1}{2}\rho S_{prop}C_{prop}\begin{bmatrix} (k_{motor}\delta_t)^2 - V_a^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -F_{Tq}^b \end{bmatrix} \end{aligned} \tag{3.42}$$

where

$$\begin{aligned} C_x(\alpha) &= -C_D(\alpha)\cos\alpha + C_L(\alpha)\sin\alpha \\ C_{x_q}(\alpha) &= -C_{D_q}\cos\alpha + C_{L_q}\sin\alpha \\ C_{x_{\delta_e}}(\alpha) &= -C_{D_{\delta_e}}\cos\alpha + C_{L_{\delta_e}}\sin\alpha \\ C_z(\alpha) &= -C_D(\alpha)\cos\alpha - C_L(\alpha)\sin\alpha \\ C_{z_q}(\alpha) &= -C_{D_q}\cos\alpha - C_{L_q}(\alpha)\sin\alpha \\ C_{z_{\delta_e}}(\alpha) &= -C_{D_{\delta_e}}\cos\alpha - C_{L_{\delta_e}}\sin\alpha \\ C_D(\alpha) &= C_{D_0} + C_{D_\alpha}\alpha \\ C_L(\alpha) &= C_{L_0} + C_{L_\alpha}\alpha \end{aligned} \tag{3.43}$$

and the external moments are given by

$$\begin{bmatrix} L^b \\ M^b \\ N^b \end{bmatrix} = \frac{1}{2}\rho V_a^2 S \begin{bmatrix} b(C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r) \\ c(C_{M_0} + C_{M_\alpha}\alpha + C_{M_q}\frac{c}{2V_a}q + C_{M_{\delta_e}}\delta_e) \\ b(C_{N_0} + C_{N_\beta}\beta + C_{N_p}\frac{b}{2V_a}p + C_{N_r}\frac{b}{2V_a}r + C_{N_{\delta_a}}\delta_a + C_{N_{\delta_r}}\delta_r) \end{bmatrix} + \begin{bmatrix} \tau_{roll} \\ \tau_{pitch} \\ \tau_{yaw} \end{bmatrix} \tag{3.44}$$

where the forces and moments generated by the quadcopter propellers are given by

$$\begin{bmatrix} F_{Tq}^b \\ \tau_{roll} \\ \tau_{pitch} \\ \tau_{yaw} \end{bmatrix} = \begin{bmatrix} C_{Tq} & (1 & 1 & 1 & 1) \\ C_{Tq}\cos(\eta) & (1 & -1 & -1 & 1) \\ C_{Tq}\sin(\eta) & (1 & 1 & -1 & -1) \\ C_{Mq} & (1 & -1 & 1 & -1) \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{3.45}$$

3.4 Estimation and Control

In this section first the state-space equation will be presented, then the definitions of observability and controllability are described. Then methods for creating observers that try to estimate the state of the system, and controllers that attempts to control the system towards a setpoint are explained. The section finishes off with describing the methods of multiple model estimation and control.

3.4.1 The State-Space Equation

In control engineering the state-space representation is a way of describing the dynamics of a system using a set of input, output and state variables that are related through first-order differential or difference equations on the form

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= A(t)\boldsymbol{x}(t) + B(t)\boldsymbol{u}(t) + L(t)\boldsymbol{\xi}(t) \\ \boldsymbol{y}(t) &= C(t)\boldsymbol{x} + D(t)\boldsymbol{u}(t) + \boldsymbol{\theta}(t) \end{aligned} \tag{3.46}$$

for continuous systems and

$$\begin{aligned} \boldsymbol{x}_{t+1} &= A(t)\boldsymbol{x}_t + B(t)\boldsymbol{u}_t L(k)\boldsymbol{\xi}_t \\ \boldsymbol{y}_t &= C(t)\boldsymbol{x}_t + D(k)\boldsymbol{u}_t + \boldsymbol{\theta}_t \end{aligned} \tag{3.47}$$

for discrete systems where $\boldsymbol{x} \in \mathbb{R}^n$ is the state of the system, $\boldsymbol{u} \in \mathbb{R}^m$ is the control input, $\boldsymbol{y} \in \mathbb{R}^p$ is the output. $\boldsymbol{\xi} \in \mathbb{R}^r$ is the process noise and $\boldsymbol{\theta} \in \mathbb{R}^q$ is the

measurement noise, and none of the noise terms can be measured. $A(t) \in \mathbb{R}^{n\times n}$ is called the state matrix, $B(t) \in \mathbb{R}^{n\times m}$ the input matrix, $C(t) \in \mathbb{R}^{p\times n}$ is the output or measurement matrix and $D(t) \in \mathbb{R}^{p\times m}$ is the direct transition matrix. In many cases, and for all systems that will be presented in this thesis D will be a zero matrix, and will be omitted from the equations. Knowledge regarding the process noise and measurement noise are given by the symmetric positive definite matrices $Q \in \mathbb{R}^{q\times q}$ and $R \in \mathbb{R}^{r\times r}$ respectively [1].

In general the system matrices are time-variant, but in many cases they can be assumed to time-invariant such that the process becomes

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k + L\boldsymbol{\xi}_k \tag{3.48}$$

$$\boldsymbol{y}_k = C\boldsymbol{x}_k + \boldsymbol{\theta}_k \tag{3.49}$$

and is now called a linear time-invariant(LTI) system.

3.4.2 Controllability

An LTI system is said to be *controllable* if and only if the *controllability matrix*

$$\mathcal{C} = [B, AB, ..., A^{n-1}B] \in \mathbb{R}^{n\times mn} \tag{3.50}$$

has full row rank [1], i.e. $\operatorname{rank}\mathcal{C} = n$. This is implies that some control input $u(t)$ exist which will take the system from an initial state x_0 to some other state x_1 within finite time. It says nothing about the time the transfer will take place, nor which trajectory it will follow. This means the control input need not be unique, it is only known that it exist. Control methods can then be designed utilising the controllability property in order to ensure that the system reaches its desired state.

3.4.3 Observability

Often the internal state of a system may be required in an application, but it may not be directly accessible. The question is then if it is possible to reconstruct the state of the system $x(t_0)$ by measuring the inputs and outputs of the system at current and future times. The answer is yes, if the system is *observable*. A system is called *observable* if the *observability matrix*

$$\mathcal{O} = [C^T, A^TC^T, ..., (A^T)^{n-1}C^T] \in \mathbb{R}^{n\times pn} \tag{3.51}$$

has full row rank, i.e. $\operatorname{rank}\mathcal{O} = n$ [1].

3.4.4 Full State Feedback Control

Given a LTI system on the form $\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}$ we wish to design a controller by feeding back the state $\boldsymbol{x}$ multiplied by a gain matrix K in such a way that it will drive the system from an initial state $\boldsymbol{x}(t_0)$ to zero by

$$\boldsymbol{u} = -K\boldsymbol{x} \tag{3.52}$$

Inserted back into the system equation yields:

$$\dot{\boldsymbol{x}} = (A - BK)\boldsymbol{x} \tag{3.53}$$

This differential equation has solution

$$\dot{\boldsymbol{x}} = e^{(A-BK)t}\boldsymbol{x}(t_0) \tag{3.54}$$

which will asymptotically go to zero as long as all eigenvalues of the matrix $(A - BK)$ lie in the left half plane [15]. Thus the problem becomes deciding the feedback gain matrix K such that this holds. It can be proven that the closed loop poles of the system can be chosen arbitrarily by an appropriate choice of K as long as the pair (A, B) is completely controllable. This means that in theory the closed loop response of the system can be chosen to follow some predefined design criterion. For lower order systems this has been investigated quite rigorously, however it becomes more and more difficult with increasing system complexity.

3.4.5 Observer Design

In the full state feedback control presented above it is assumed that the full state of the system is available for feedback to make up the controller. This is most often not possible due to practical reasons, because it may not be feasible to have sensors which measure every element of the state. Then it would be nice if it was possible to make an estimate of the part of the state that is not measured directly. A full state observer is given by [1]

$$\dot{\hat{\boldsymbol{x}}} = A\hat{\boldsymbol{x}} + B\boldsymbol{u} + L(\boldsymbol{y} - C\hat{\boldsymbol{x}}) \tag{3.55}$$

where $\hat{\boldsymbol{x}}$ is the estimate of the system state $\boldsymbol{x}$. The observer design procedure is then the process of determining the observer gain matrix L in such a way that the estimation error $\boldsymbol{e}(t) = \boldsymbol{x} - \hat{\boldsymbol{x}}$ asymptotically goes to zero, i.e. $\boldsymbol{e}(t) \to \boldsymbol{0}$ as $t \to \infty$. Taking the time derivative of the error and substituting yields

$$\begin{aligned}\dot{\boldsymbol{e}} =& \dot{\boldsymbol{x}} - \dot{\hat{\boldsymbol{x}}} \\ =& A\boldsymbol{x} + B\boldsymbol{u} - (A\hat{\boldsymbol{x}} + B\boldsymbol{u} + L(\boldsymbol{y} - C\hat{\boldsymbol{x}})) \\ =& (A - LC)\boldsymbol{e}\end{aligned} \tag{3.56}$$

For the observant reader this looks oddly familiar to the equation obtained when making the state feedback controller, and it is. The observer gain matrix L can guarantee that the estimation tracking error will asymptotically go to zero as long as the eigenvalues of the matrix $(A - LC)$ all are in the left half-plane. The matrix L can be used to arbitrarily determine these eigenvalues as long as the pair (A, C) is completely observable.

3.4.6 State Compensator

Combining the methods of state estimation through a full state observer, and a full state feedback controller the resulting systems is called a state compensator [15]. Using the estimate in the feedback controller $\boldsymbol{u} = -K\hat{\boldsymbol{x}}$ into the system equation gives

$$\begin{aligned}\dot{\boldsymbol{x}} =& A\boldsymbol{x} + B\boldsymbol{u} \\ =& A\boldsymbol{x} - BK\hat{\boldsymbol{x}} \\ =& (A - BK)\boldsymbol{x} + BK\boldsymbol{e}\end{aligned} \tag{3.57}$$

Combining this with the differential equation derived for the observer tracking error yields

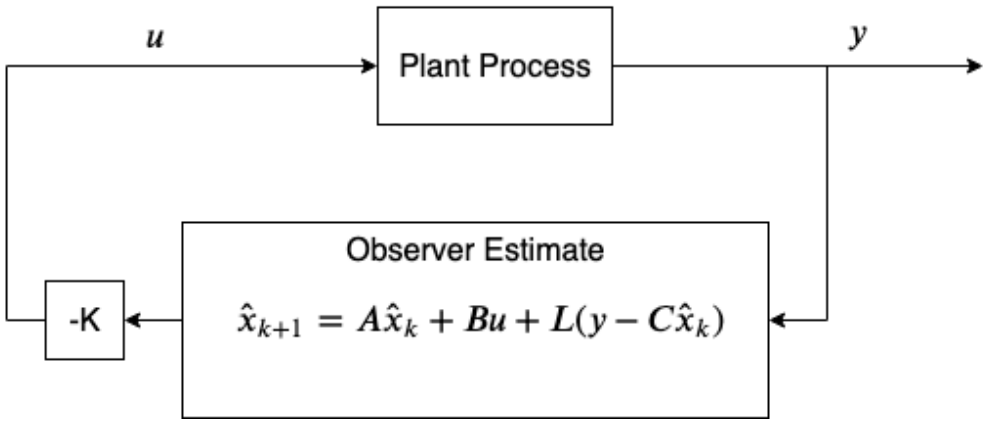

Figure 3.4: Diagram of the control loop with state estimator and linear full state feedback control.

$$\begin{bmatrix}\dot{\boldsymbol{x}} \\ \boldsymbol{e}\end{bmatrix} = \begin{bmatrix}A - BK, & BK \\ 0, & A - LC\end{bmatrix}\begin{bmatrix}\boldsymbol{x} \\ \boldsymbol{e}\end{bmatrix} \tag{3.58}$$

From this it can be derived that the system state $\boldsymbol{x}(t)$ asymptotically will go to zero and that the observer tracing error tracking will go to zero as long as the eigenvalues of the matrix in Equation (3.58) are in the left half-plane. This is ensured by the methods outlined earlier even if the feedback gain and observer gain were derived independent of each other. Figure (3.4) shows a diagram illustrating the usual control loop when combining state estimation with linear full state feedback control.

3.4.7 LQR Control

Optimal control is a part of control theory which concerns itself with finding a control for a given system in order to minimize a specified optimality criterion(cost function). Within optimal control there is a subspace of methods concerned with finding the optimal state feedback control law of the form $\boldsymbol{u} = K\boldsymbol{x}$ for a linear system given a quadratic cost function. This is called a linear quadratic regulator (LQR) [1, 46]. A brief outline of how to obtain the infinite horizon control law (the feedback gain is constant) for a discrete LTI system will be outline here.

Given an LTI system of the form of Equation (3.48) we wish to determine the control sequence $\boldsymbol{u}_k^*$, $k \geq 0$ which minimises the quadratic cost function

$$J(\boldsymbol{u}) = \sum_{k}^{\infty} [\boldsymbol{x}_k^T Q_{LQR} \boldsymbol{x}_k + \boldsymbol{u}_k^T R_{LQR} \boldsymbol{u}_k] \tag{3.59}$$

for any initial state $\boldsymbol{x}_0$, where the weighting matrices Q_{LQR} and R_{LQR} are real, symmetric and positive definite.

Assuming that the LTI state-space system is controllable and observable it can be shown that the solution to the LQR problem is given by the linear state-feedback control law

$$\boldsymbol{u}_k^* = K\boldsymbol{x}_k = -[R + B^T P_c^* B]^{-1} B^T P_c^* A \boldsymbol{x}_k \tag{3.60}$$

where P_c^* is the solution to the algebraic Riccati equation

$$P_c = A^T P_c A - A^T P_c B [R + B^T P_c B]^{-1} B^T P_c A + Q \tag{3.61}$$

Deriving the infinite horizon LQR control for a continuous time time-invariant system can be done in an analogous way. Derivations for time-variant systems and finite-time solutions can be found in the abundant literature regarding optimal control and LQR control available.

3.4.8 Integral Action

A state feedback controller is only ensured to reach the desired reference point $\boldsymbol{x}^d(t)$ as long as the plant which is going to be controlled is perfectly modelled. If there are uncertainties in the process parameters, disturbances or unmodelled dynamics there may be steady state errors that the linear state feedback controller is not able to eliminate on its own. The regular state feedback controller only applies a gain based on the error state at the current time. It will not be able to identify and correct errors that have accumulated over time in the system, or dynamics that have not been modelled.

The basic approach to solving this is to augment the controller with an integral state $\boldsymbol{z}(t) = \int (\boldsymbol{x}^d(t) - \boldsymbol{x}(t))\, dt$ whose dynamics are determined by $\dot{\boldsymbol{z}} = \boldsymbol{x}^d(t) - \boldsymbol{x}(t)$. One method is to extend the control state feedback controller to $\boldsymbol{u} = -K(\boldsymbol{x} - \boldsymbol{x}_d) + K_I \boldsymbol{z}$, where K_I is a hand-tuned integral gain. With this, however, the optimality properties obtained by the LQR controller is lost.

In order to maintain the optimality properties, another approach is to include the integral state in the LQR gain computation. Extending the state vector with the integral state, and augmenting the state-space matrices yields and augmented state-space system given by

$$\begin{aligned} \tilde{\boldsymbol{x}} &= \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{bmatrix}, \quad \widetilde{A} = \begin{bmatrix} A & 0 \\ -I & 0 \end{bmatrix} \\ \widetilde{B} &= \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad \widetilde{C} = \begin{bmatrix} C & 0 \end{bmatrix}. \end{aligned} \tag{3.62}$$

This augmented state-space system can then be used to compute the state feedback LQR controller gain in the usual manner [28].

3.4.9 Adaptive Estimation and Control

The following sections will describe how to construct adaptive estimators and controllers based on multiple model adaptive (MMA) theory, and linear parameter-varying (LPV) theory utilising the state-space concepts described in the previous sections.

3.4.10 Multiple Model Adaptive Estimation

To make a state observer for a dynamical system, exact parameter values for the plant is required in order to ensure good performance. These parameter values may be difficult to determine, and bad estimates may lead to poor or even fatal performance. In order to cope with these uncertainties a wide array of methods have been developed in order to make a system that is able to adapt to changes or uncertainties. One of these methods is Multiple Model Adaptive

Estimation (MMAE). In short MMAE creates several proposed models using various values of the uncertain parameters. Estimation filters are then set up for each proposed model, and conditional probabilities are created for each estimated value by use of some performance metric compared to the real measured values [6, 17]. Figure (3.5) shows a diagram illustrating the general setup of the MMAE algorithm.

The initial research done using MMAE made use of the Kalman filter for the local state estimates. In this thesis the method of using steady-state Krener observers together with a minimum energy criterion developed by Hassini et. al. [17] will be used instead.

Consider a multiple-input-multiple-output LTI plant of the form

$$\begin{aligned} \boldsymbol{x}_{t+1} &= A(\kappa)\boldsymbol{x}_t + B(\kappa)\boldsymbol{u}_t + G(\kappa)\boldsymbol{\xi}_t \\ \boldsymbol{y}_t &= C(\kappa)\boldsymbol{x}_t + \boldsymbol{\theta}_t \end{aligned} \tag{3.63}$$

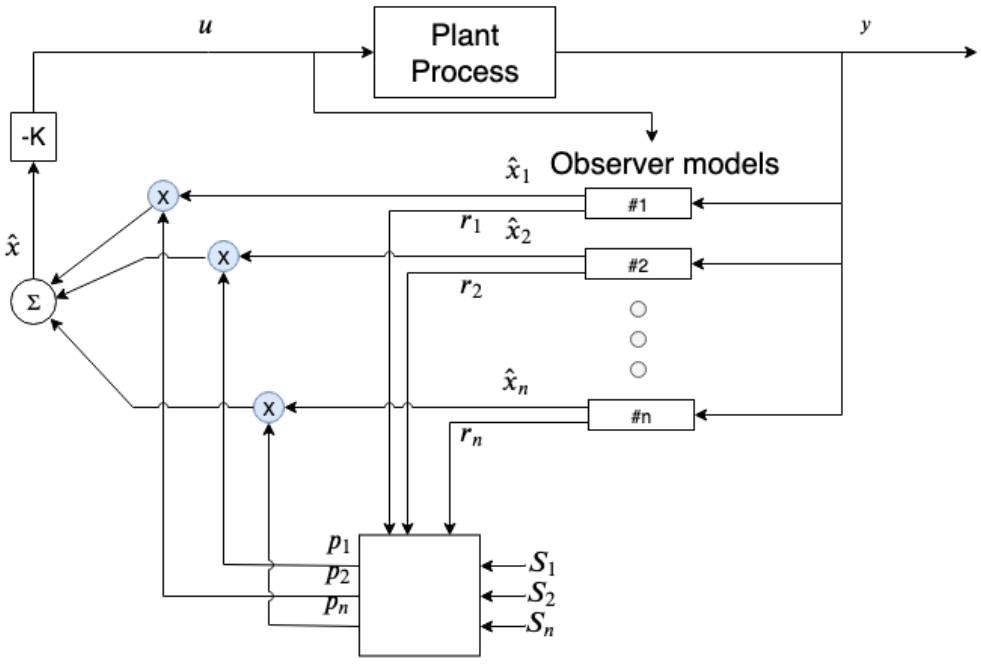

Figure 3.5: Diagram of the multiple model adaptive estimation algorithm.

where ξ_t and θ_t are unknown deterministic plant and measurement noises respectively. $\kappa \in \mathbb{R}^l$ are the uncertain parameters in the plant. Consider now creating a set of N candidate models using various possible realisations κ_i of the uncertain value κ. For each of these local models a local state estimate based on the Krener observer can be designed

$$\begin{aligned}\hat{\boldsymbol{x}}_{t+1}^{\kappa_i} &= A_{\kappa_i}\hat{\boldsymbol{x}}_t^{\kappa_i} + B_{\kappa_i}\boldsymbol{u}_t + H_{\kappa_i}(\boldsymbol{y}_t - C_{\kappa_i}\hat{\boldsymbol{x}}_t^{\kappa_i}) \\ \hat{\boldsymbol{y}}_t^{\kappa_i} &= C_{\kappa_i}\hat{\boldsymbol{x}}_t^{\kappa_i} \\ H_{\kappa_i} &= \Sigma_{\kappa_i}C_{\kappa_i}^T[C_{\kappa_i}\Sigma_{\kappa_i}C_{\kappa_i}^T + \Theta]^{-1}\end{aligned} \tag{3.64}$$

where Σ_{κ_i} is the solution of the steady-state discrete Riccati equation

$$\begin{aligned}0 = &-\Sigma_{\kappa_i} + A_{\kappa_i}\Sigma_{\kappa_i}A_{\kappa_i}^T + G_{\kappa_i}\Gamma G_{\kappa_i}^T \\ &- A_{\kappa_i}^T\Sigma_{\kappa_i}C_{\kappa_i}^T[C_{\kappa_i}\Sigma_{\kappa_i}C_{\kappa_i}^T + \Theta]^{-1}C_{\kappa_i}\Sigma_{\kappa_i}A_{\kappa_i}.\end{aligned} \tag{3.65}$$

Where the subscript κ_i denotes a realisation of the model using that specific candidate parameter value for κ. It is assumed that the pairs $[A_{\kappa_i}, B_{\kappa_i}]$ and $[A_{\kappa_i}, C_{\kappa_i}]$ are completely controllable and observable, respectively. Γ and Θ are symmetric positive definite weighting matrices that contain information regarding the process and measurement noise. A state estimate can thus be given by

$$\begin{aligned}\hat{\boldsymbol{x}}_t &= \sum_{i=1}^{N} p_t^i \hat{\boldsymbol{x}}_t^{\kappa_i} \\ \hat{\boldsymbol{y}}_t &= \sum_{i=1}^{N} p_t^i \hat{\boldsymbol{y}}_t^{\kappa_i}\end{aligned} \tag{3.66}$$

where p_t^i are dynamic weights that weight the various models against each other. It is given by

$$p_{t+1}^i = \frac{\beta_i e^{-w_t^i}}{\sum_{j=1}^{N}\beta_j e^{-w_t^j}} p_t^i \tag{3.67}$$

where β_i is a positive constant weighting and w_t^i is an *error measuring function* or goodness measure that maps the real measured signal and the estimated output of the system to a real positive value. The goodness measure and weights that were used in this thesis are given by

$$w_t^i = \frac{1}{2}\|\boldsymbol{r}_t^{\kappa_i}\|_{S_{\kappa_i}^{-1}}^2, \qquad \beta_i = \frac{1}{\sqrt{|S_{\kappa_i}|}} \tag{3.68}$$

where $\boldsymbol{r}_t^{\kappa_i} = \boldsymbol{y}_t - \hat{\boldsymbol{y}}_t^{\kappa_i}$, and $\|\boldsymbol{x}\|_S = \sqrt{\boldsymbol{x}^T S \boldsymbol{x}}$. S_{κ_i} are weighting matrices given by

$$S_{\kappa_i} = C_{\kappa_i}\Sigma_{\kappa_i}C_{\kappa_i}^T + \Theta, \tag{3.69}$$

which is important in order to scale the estimation errors for them to be comparable.

It will not be explained in detail here, but it can be proved that if the initial conditions of the dynamic weights satisfy $p_0^i \in (0,1)$ and $\sum p_0^i = 1$, they will be restricted to be within the interval $[0,1]$ and satisfy $\sum p_t^i = 1$ for all $t \geq 0$. The dynamic weights can then more intuitively be viewed as the probability that a given candidate model is the correct one. Furthermore it can be proven that the MMAE method will converge towards the candidate model which results in the least output error measure w_t^i, i.e. the system will converge towards the model with parameter value κ_{i^*} that is the most similar to the true parameter value.

3.4.11 Multiple Model Adaptive Control

In the control side of things there exist similar methods with the use of multiple models in order to control the system, named Multiple-Model Adaptive Control(MMAC). In the realm of MMAE the objective is only to correctly identify the candidate model with parameter values closest to the true parameter values of the plant given some similarity criterion by passively observing the outputs of the system, and giving out the optimal estimate of the state of the system. Controllers on the other hand will also have to be able to ensure the stability of the system while simultaneously switching between the different models.

The design of multiple model adaptive controllers, and how to switch between models while also ensuring that the control error tends to zero. The general gist of things is that the system contains N identification models denoted $\{I_j\}$ operating in parallel. Each identification model has a corresponding parameterized controller C_j of parameters ζ_j which are chosen such that C_j achieves the control objectives of I_j. At every time instant, each identification model produces an output $\boldsymbol{y}_j$ and the corresponding controller produces a control input $\boldsymbol{u}_j$. The task of the MMAC algorithm is then at each time step to dictate which control input to use for controlling the plant [6, 30].

Both the identification model I_j and corresponding controller C_j might be either linear, non-linear, static or adaptive. The general idea when deciding on which model and controller to use is to determine the performance of each model according to some performance cost index $J_j(t)$ and choose the one with the lowest cost index. In the work by Narendra et. al. [30] several techniques are outlined describing how stability can be proved of both adaptive and fixed models even while running the risk of selecting a controller that might render the system unstable.

In this thesis a simpler approach will be taken based on some of the earlier work on multiple-model adaptive control such as [7]. Instead of explicitly choosing and switching between the proposed candidate models at each time step, the performance cost index will be used to directly compute a convex combination of the control input form each of the candidate models to compute the resulting control input to send to the plant.

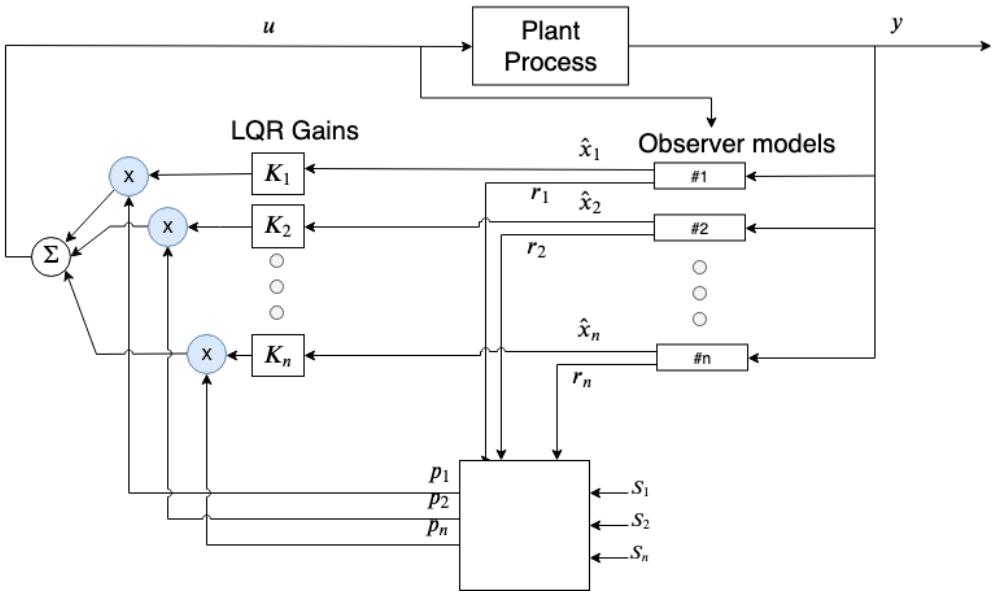

Figure 3.6: Diagram of the MMA algorithm extended with full feedback control gains for each of the candidate models

Given the candidate models of Equation (3.63) a corresponding LQR controller gain K_{κ_j} can be compute for each of them given by Equation (3.60) as illustrated in Figure (3.6). Utilising the model goodness measure of Equation (3.68) as the performance cost index, the dynamic weighting of each model can be computed using Equation (3.67). Thus utilising the properties of the dynamic weights ($\{\sum p^j = 1, p^j \in [0,1] \forall\, p^j\}$) a linear combination of the control inputs can be computed to give the resulting control input

$$\boldsymbol{u}_t = \sum_i^N p_t^i \boldsymbol{u}_t^i \tag{3.70}$$

3.4.12 Linear Parameter Varying Systems

Linear parameter-varying(LPV) systems are linear dynamical systems whose state-space representation depend on exogenous non-stationary parameters $\boldsymbol{v}(k)$ [12, 42]. Here LPV systems on forms similar to the state-space systems of Equation (3.47) will be considered, such as

$$\begin{aligned} \boldsymbol{x}_{t+1} &= A(\boldsymbol{v}(k))\boldsymbol{x}_t + B(\boldsymbol{v}(t))\boldsymbol{u}_t G(\boldsymbol{v}(t))\boldsymbol{\xi}_t \\ \boldsymbol{y}_t &= C(\boldsymbol{v}(t))\boldsymbol{x}_t + \boldsymbol{\theta}_t \end{aligned} \tag{3.71}$$

The parameters $\boldsymbol{v}(t)$ are generally assumed to be bounded and able to take values within some convex polytope(e.g. hyper-rectangle) $\Delta\boldsymbol{v}$, and they

are measured or readily available at run-time. If the LPV system is to be used as an approximation of a non-linear system a common approach is to capture the non-linearities in the state variables and use $\boldsymbol{v}(t)$ as a scheduling parameter in order to imitate linearity. It is then called a quasi-linear parameter-varying(qLPV) system.

An LPV system is called polytopic if the dependence of the state-space matrices in Equation (3.71) on $\boldsymbol{v}(k)$ is affine, and the values of $\boldsymbol{v}(t)$ ranges over the polytope $\Delta\boldsymbol{v}$ [37]. Thus, the polytopic LPV system can be described as a convex sum of L constant vertex systems (A^j, B^j, C^j, G^j)

$$(A(\boldsymbol{v}(t)), B(\boldsymbol{v}(t)), C(\boldsymbol{v}(t), G(\boldsymbol{v}(t)) = \sum_{j=1}^{L} h^j(\boldsymbol{v}(k))(A^j, B^j, C^j, G^j) \tag{3.72}$$

where the non-negative coefficients $h^j(\boldsymbol{v}(t))$ have the following properties

$$\sum_{j=1}^{L} h^j(\boldsymbol{v}(t)) = 1, \qquad h^j(\boldsymbol{v}(t)) \in [0,1], \qquad \forall \boldsymbol{v} \in \Delta\boldsymbol{v} \tag{3.73}$$

3.4.12.1 LPV State Estimation

Given a polytopic LPV system on the form of Equation (3.71) a state estimator can be obtained through the LPV Kalman filter [43]:

$$\begin{aligned} \hat{\boldsymbol{x}}_t &= \sum_{j=1}^{L} \hat{\boldsymbol{x}}_t^j \\ \hat{\boldsymbol{y}}_t &= C(\boldsymbol{v}(t))\hat{\boldsymbol{x}}_t \end{aligned} \tag{3.74}$$

where the estimates $\hat{\boldsymbol{x}}_t$ are obtained by

$$\begin{aligned} \hat{\boldsymbol{x}}_t^j &= (I - H^j C^j)\hat{\boldsymbol{x}}_t^{-j} + K^j h^j(\boldsymbol{v}(t))\boldsymbol{y}_t \\ \hat{\boldsymbol{x}}_{t+1}^{-j} &= A^j \hat{\boldsymbol{x}}_t^j + h^j(\boldsymbol{v}(t))B^j \boldsymbol{u}_t. \end{aligned} \tag{3.75}$$

The superscript $(-)$ indicates predictions made by the state estimator before the measurement is taken into account. The Kalman filter gain H^j of vertex system j is given by

$$H^j = P^j (C^j)^T (C^j P^j (C^j)^T + R)^{-1}, \tag{3.76}$$

where P^j is the solution of the discrete Riccati equation

$$P^j = A^j P^j (A^j)^T - A^j H^j C^j P^j (A^j)^T + G^j Q (G^j)^T. \tag{3.77}$$

3.4.13 Multiple Model Linear Parameter Varying Adaptive Estimation and Control

Combining the ideas of the multiple models techniques together with the linear parameter-varying methods an architecture designed for analysis and control of non-linear systems (LPV part) while simultaneously adapt to parameter uncertainties in the plant (MMAE part) is arrived at [36].

From the basis of the theory stated earlier the method is pretty straight forward. Assuming that there are uncertain parameters $\boldsymbol{\kappa}$ in the plant a set of N candidate models are constructed based on a discrete set of realisations $\boldsymbol{\kappa}_i$ of the uncertain parameters $\boldsymbol{\kappa}$. For a non-linear plant each candidate model can be brought to a quasi-LPV form where the parameters $\boldsymbol{v}(t)$ depend on the state, input and outputs of the system. Assuming that the state-space matrices for each of the candidate models are polytopic local state estimates can be obtained for each candidate model from the LPV Kalman filter:

$$\begin{aligned}
\hat{\boldsymbol{x}}_t &= \sum_{i=1}^{N} p_t^i \hat{\boldsymbol{x}}_t^j(\kappa_i) \\
\hat{\boldsymbol{y}}_t &= \sum_{i=1}^{N} p_t^i \hat{\boldsymbol{y}}_t^j(\kappa_i) \\
\hat{\boldsymbol{x}}_t^j(\kappa_i) &= \sum_{j=1}^{L} \hat{\boldsymbol{y}}_t^j(\kappa_i) \\
\hat{\boldsymbol{y}}_t(\kappa_i) &= C_{\kappa_i}^j(\boldsymbol{v}(t))\hat{\boldsymbol{x}}_t^j(\kappa_i)
\end{aligned} \tag{3.78}$$

where each of the local Kalman filter estimates are given by

$$\begin{aligned}
\hat{\boldsymbol{x}}_t^j(\kappa_i) &= (I - H_{\kappa_i}^j C_{\kappa_i}^j)\hat{\boldsymbol{x}}_t^{-j}(\kappa_i) + K_{\kappa_i}^j h_{\kappa_i}^j(\boldsymbol{v}(t))\boldsymbol{y}_t \\
\hat{\boldsymbol{x}}_{t+1}^{-j}(\kappa_i) &= A_{\kappa_i}^j \hat{\boldsymbol{x}}_t^j(\kappa_i) + h_{\kappa_i}^j(\boldsymbol{v}(t))B_{\kappa_i}^j \boldsymbol{u}_t
\end{aligned} \tag{3.79}$$

with Kalman filter gain

$$H_{\kappa_i}^j = P_{\kappa_i}^j (C_{\kappa_i}^j)^T (C_{\kappa_i}^j P_{\kappa_i}^j (C_{\kappa_i}^j)^T + R)^{-1} \tag{3.80}$$

and $P_{\kappa_i}^j$ given by the solutions to the discrete Riccati equation

$$P_{\kappa_i}^j = A_{\kappa_i}^j P_{\kappa_i}^j (A_{\kappa_i}^j)^T - A_{\kappa_i}^j H_{\kappa_i}^j C_{\kappa_i}^j P_{\kappa_i}^j (A_{\kappa_i}^j)^T + G_{\kappa_i}^j Q (G_{\kappa_i}^j)^T. \tag{3.81}$$

The dynamic weights p_t^j are given in the same way as in Equation(3.67). The error measuring function ω_t^i and matrix weighting β_t^i will now be given by

$$w_t^i = \frac{1}{2}\|\boldsymbol{r}_t^{\kappa_i}\|^2_{S_{\kappa_i}^{-1}()\boldsymbol{v}(t)'} \qquad \beta_i = \frac{1}{\sqrt{|S_{\kappa_i}(\boldsymbol{v}(t))|}} \tag{3.82}$$

with $\boldsymbol{r}_t^{\kappa_i} = \boldsymbol{y}_t - \hat{\boldsymbol{y}}_t(\kappa_i)$ and $S_{\kappa_i}(\boldsymbol{v}(t)$ chosen to be

$$\begin{aligned} S_{\kappa_i}(\boldsymbol{v}(t) &= \sum_{j=1}^{L} h^j(\boldsymbol{v}(t)) S_{\kappa_i}^j \\ S_{\kappa_i}^j &= C_{\kappa_i}^j P_{\kappa_i}^j (C_{\kappa_i}^j)^T + R. \end{aligned} \tag{3.83}$$

3.4.13.1 LPV Multiple Model Adaptive Control

Given that the LPV systems for all the candidate models are polytopic each of the vertex systems will be time-invariant. The proposed method for controlling the system is then to design an LQR controller gain $K_{\kappa_i}^j$for each of the vertex systems and setting the resulting control input arising from the candidate model as the linear combination of the control input from the vertex systems weighted by the coefficients $h^j(\boldsymbol{v}(t))$

$$\begin{aligned} \boldsymbol{u}_t &= \sum_{i}^{N} p_t^i \boldsymbol{u}_t(\kappa_i) \\ \boldsymbol{u}_t(\kappa_i) &= \sum_{j}^{L} h^j(\boldsymbol{v}(t)) \boldsymbol{u}_t^j(\kappa_i) = -\sum_{j}^{L} h^j(\boldsymbol{v}(t)) K_{\kappa_i}^j \hat{\boldsymbol{x}}_t. \end{aligned} \tag{3.84}$$

Chapter 4

Simulation

This chapter contains descriptions of the models that were utilised for simulations based on theory described in the previous chapter. Approaches for model linearisations and controller design will also be described. For all simulations the plant was a direct implementation of the differential equations of the dynamics and solved by explicit Euler integration. The matrices in the state-space systems used for observer design and controllers were obtained by Euler discretisation.

4.1 Single Axis Simulation

In order to verify the functionality of the MAA algorithm and the code, the first simulations were performed on a simplified model inspired by the models presented in [21]. It was chosen to use the model for pitch and x dynamics (Equation 12) from that paper.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \epsilon \\ 0 \\ 1 \end{bmatrix} u \tag{4.1}$$

This is the system after a coordinate transformation $\boldsymbol{x} = [x, \dot{x}, -g\theta, -g\dot{\theta}]^T$, $u = -g\tau_{pitch}/I_{yy}$ and $\epsilon = -I_{yy}/mgL$ where L is a system parameter for this model. This is the linearised model for a type of ducted fan tailsitter UAV. This model was chosen since it is single-input-single-output and is still within the UAV genre, and a bit more interesting than the linearised quadcopter model. For the simulations it was assumed that the parameter ϵ was unknown and needed to be determined through MMAE. The same linearised model in Equation (4.1) with $\epsilon = 0.45$ was used as the plant model. 5 candidate observer models were used it was assumed that ϵ was within the range [0.05, 1]. A full state feedback controller $u = -K(\hat{\boldsymbol{x}} - \boldsymbol{x}_d)$ was used, where $\boldsymbol{x}_d$ denotes a

desired set-point. The feedback gain $K = [k_1, k_2, k_3, k_4]$ was manually tuned to values that gave a satisfactory system response. According to [21] the system is stable as independent of the specific value of ϵ as long as k_2 and k_3 are large enough compared to ϵk_1. With this, the feedback gain was $K = [5, 25, 50, 50]$. The simulations were done without any added process or measurement noise. The initial state was $\boldsymbol{x}_0 = [0, 0, 0.3, 0]^T$, and the system was set to hover around a desired position $\boldsymbol{x}_d = [1 + 0.1sin(0.1t), 0, 0, 0]^T$. The sine wave was put into the system to ensure persistent excitation and avoid information death [5]. The dynamic weights were capped at 0.95, and the residue was distributed among the other weights in order to maintain $\sum p_i = 1$. This was done in order to ensure that the system was able to swap between which estimates it chose to use. If the dynamic weight of any of the candidate models go to zero it can easily be seen from Equation (3.67) that the system would never have been able to increase it again in order to change model.

4.2 Simulation of Six DoF Quadcopter Model

The next step was to simulate a quadcopter during hover while using the full six degrees of freedom kinematic and dynamic models.

4.2.1 Nonlinear Simulation Model

The plant was modelled as a quadcopter, and no aerodynamic effects were considered. This means that the only external forces and moments were gravity and the control moments from the propellers, resulting in the external forces being represented by.

$$\begin{bmatrix} F_x^b \\ F_y^b \\ F_z^b \end{bmatrix} = \begin{bmatrix} -g\sin(\theta) \\ g\sin(\phi)\cos(\theta) \\ g\cos(\phi)\cos(\theta) \end{bmatrix} + \frac{1}{m}\begin{bmatrix} 0 \\ 0 \\ F_T \end{bmatrix} \tag{4.2}$$

and the external moments in Equation (3.44) became the control commands

$$\begin{bmatrix} L^b \\ M^b \\ N^b \end{bmatrix} = \begin{bmatrix} \tau_{roll} \\ \tau_{pitch} \\ \tau_{yaw} \end{bmatrix} \tag{4.3}$$

Without any specific model to adhere to the quadcopter was modelled as a $1m \times 1m \times 0.2m$ box. The box was set to weigh $10kg$ with a uniform weight distribution. This resulted in a diagonal inertia matrix J_{box} for the quadcopter body. Fixed-wing aircraft on the other hand do not have a symmetric shape along the x_b-axis and uniform weight distribution. In addition, a new payload might be fitted onto the Kestrel. To simulate this, an additional point mass

was added to the aircraft at a position $[x_p^b, y_p^b, z_p^b]$ relative to the UAV CM. This resulted in an addition J_{point} to the inertia matrix given by Equation(3.14) and the total inertia matrix can be found by superposition of the two contributions $J_{tot} = J_{box} + J_{point}$.

4.2.2 Linearising the Model

Equations (3.38-3.41) with external forces and moments given by Equations (4.2,4.3) will be used to simulate the full six DoF movement of the quadcopter. However those equations are highly coupled and non-linear and are not well suited for linear observer and controller design. It is therefore customary to linearise the equations around the desired operating condition [8, 40].

If it is assumed that the roll ϕ and pitch θ will be small Equation (3.40) can be simplified as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.4}$$

Similarly, assuming that the angular rates $[p, q, r]$ are small the body angular accelerations of Equation (3.41) can be simplified to

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{J_{zz}}{\Gamma} & 0 & \frac{J_{xz}}{\Gamma} \\ 0 & \frac{1}{\Gamma} & 0 \\ \frac{J_{xz}}{\Gamma} & 0 & \frac{J_{xx}}{\Gamma} \end{bmatrix} \begin{bmatrix} L^b \\ M^b \\ N^b \end{bmatrix} \tag{4.5}$$

The linear dynamics might however be a bit trickier to linearise. Depending on the goals for the control designs and the movement of the UAV different approaches might be taken. Also, in the case of other external forces than gravity and quadcopter propeller thrust, some of the simplifications encountered in various other literature might not be applicable. However, for the most part it may be assumed that the roll ϕ and pitch θ angles are small, and that the angular velocities and accelerations are small. This means that the Coriolis forces in Equation (3.39) can be neglected resulting in

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_x^b \\ F_y^b \\ F_z^b \end{bmatrix} \tag{4.6}$$

assuming that it is sufficient for the aircraft to take off vertically and keep its heading angle fixed ($\psi = 0$).

4.2.3 Controlling the Quadcopter

To control the quadcopter an infinite horizon LQR controller was used with and additional integral term on altitude control in order to deal with the offset generated by gravity. Due to the model specific nature of the LQR controller each candidate model had its own feedback gain and the resulting control input became a linear combination of all the control inputs from each of the N models and the dynamic weights of each model

$$\boldsymbol{u}_k = \sum_j^N p_k^j K_j (\boldsymbol{x}_{d,k} - \hat{\boldsymbol{x}}_k^j) + K_I \boldsymbol{x}_{I,k} \tag{4.7}$$

with $K_I = [100, 10]$ and $\boldsymbol{x}_{I,k} = \sum_{t=0}^{k} (\boldsymbol{x}_{d,k} - \hat{\boldsymbol{x}}_k^j)\, \Delta_t$

To create the LQR gains weighting matrices

$$Q_{LQR} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

and

$$R_{LQR} = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \tag{4.9}$$

were used.

4.3 Longitudinal Aerodynamic Simulation

The full 12 non-linear full six DoF dynamic equations for the full UAV is difficult to design controllers for. In aircraft control it is therefore normal to decouple the linearised equations of motion into two systems that are individually more tractable for controller design [35]. These two systems are

the altitude-longitudinal system and the lateral system. The longitudinal will concern itself with the forces and moments along the longitudinal movement of the UAV that is x_b, z_b and θ. The lateral system will then concern itself with forces and moments along y_b, ϕ and ψ. The linearisations will be withing parameter ranges where the coupling effects can be assumed to be small and can be handled by disturbance rejection in the controller. In this thesis only a simulation and control of the longitudinal system will be considered. During the transition phase from quadcopter to fixed-wing mode the UAV will in most cases go in a straight line from a standstill state with no aerodynamic lift, up to trim velocity where all lift is generated from the wings.

4.3.1 Simulation Model

Considering a movement purely in the $x_b z_b$-plane, and assuming that roll ϕ and yaw ψ will be zero, Equation (3.38) becomes

$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{bmatrix} = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{4.10}$$

Next, assuming that there will no forces and thus movement in the lateral direction i.e. setting velocity v in the y_b-direction to zero, as well as assuming that roll moment p and yaw moment r will be zero, then Equation (3.39) becomes

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -qw \\ 0 \\ qu \end{bmatrix} + \frac{1}{m} \begin{bmatrix} F_x \\ 0 \\ F_z \end{bmatrix} \tag{4.11}$$

Using the same assumptions as above Equation (3.40) becomes

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_\theta \\ 0 & 1 & 0 \\ 0 & 0\frac{1}{c_\theta} & \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.12}$$

and with no lateral moments Equation (3.41) turns into

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{J_{yy}} M \\ 0 \end{bmatrix} \tag{4.13}$$

Considering only the components of lateral movement yields a system state described by $\boldsymbol{x} = [p_n, p_d, u, w, \theta, q]^T$

$$
\begin{aligned}
\dot{p}_n &= c_\theta p_n + s_\theta p_d \\
\dot{p}_n &= -s_\theta p_n + c_\theta p_d \\
\dot{u} &= -qw + \frac{1}{m} F_x \\
\dot{w} &= qu + \frac{1}{m} F_z \\
\dot{\theta} &= q \\
\dot{q} &= \frac{1}{J_{yy}} M
\end{aligned}
\tag{4.14}
$$

The forces and moments $[F_x, F_z, M]$ can be substituted from Equations (3.42) and (3.44). Using $\beta = \phi = r = p = v = 0$ yield

$$
\begin{aligned}
F_x &= -mg\sin(\theta) + \frac{1}{2}\rho V_a^2 S[C_X(\alpha) + \frac{c}{2V_a} C_{X_q} q + C_{X_{\delta_e}}(\alpha)\delta_e] + F_{Tr} \\
F_z &= mg\cos\theta + \frac{1}{2}\rho V_a^2 S[C_Z(\alpha) + \frac{c}{2V_a} C_{Z_q} q + C_{Z_{\delta_e}}(\alpha)\delta_e] + F_T \\
M &= \frac{1}{2}\rho V_a^2 c S[C_{M_0} + C_{M_\alpha}\alpha + C_{M_q} q] + \tau_{pitch}
\end{aligned}
\tag{4.15}
$$

4.3.2 Aircraft Model

With no complete list of aerodynamic coefficients for the Kestrel UAV it was decided to use the parameters for the Aerosonde UAV [35]. Aerosonde is a 13.5 kg and 2.8956 m wingspan fixed-wing UAV. The biggest difference from the Kestrel design-wise is that the forward thrust propeller is mounted at the read of the fuselage instead of in front. The necessary parameters required for modelling of the longitudinal flight are given in Table (4.1).

Name	Value
m	$13.5kg$
J_{yy}	$1.135kg/m^2$
ρ	$1.2682\ kg/m^3$
c	$0.18994m$
S	$0.55m^2$
S_{prop}	$0.2027m^2$
C_{prop}	1.0
k_{motor}	80
C_{L_0}	0.28
C_{D_0}	0.03
C_{M_0}	-0.02338
C_{L_α}	3.45
C_{D_α}	0.30
C_{M_α}	-0.38
C_{L_q}	0.0
C_{D_q}	0.0
C_{M_q}	-3.6
$C_{L_{\delta_e}}$	-0.36
$C_{D_{\delta_e}}$	0.0
$C_{M_{\delta_e}}$	-0.5

Table 4.1: Aerodynamic parameters for the Aerosone UAV required for simulation of longitudinal flight [35].

4.3.3 State-Space Observer and Controller Design

In this section, methods for generating state-space systems for the longitudinal equations of motion for a fixed-wing aircraft will be outlined.

4.3.3.1 Trim Conditions

It is comparatively difficult to create LTI state-space systems on the form of Equation (3.47) for a fixed-wing aircraft versus a quadcopter. The quadcopter in this case was only required to hover and perform small movements and thus a relatively good linearised model could be achieved by linearising the model around **0** for Euler-angles and angular rates. Fixed-wing aircraft on the other

hand require quite a substantial airspeed $\boldsymbol{V_a}$ across the wings and a positive angle-of-attack α in order to generate lift.

A non-linear system can be described in the short form $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$, where $\boldsymbol{x}$ is the system state and $\boldsymbol{u}$ is the input vector. The system is in equilibrium at the state $\boldsymbol{x}^*$ and input $\boldsymbol{u}^*$ when $f(\boldsymbol{x}^*, \boldsymbol{u}^*) = \boldsymbol{0}$. During steady flight a subset of states of the UAV will be in equilibrium. Often this will be the altitude p_n, body frame velocities $[u, v, w]$, Euler angles $[\phi, \theta, \psi]$ and body angular rates $[p, q, r]$. When this occurs the UAV is in trim [35].

4.3.3.2 Model Linearisation

Now, given that a set of trim conditions $\boldsymbol{x}^*, \boldsymbol{u}^*$ have been found, a linearised model can be found by linearising the system around the obtained trim point. The deviation from the trim state can be expressed by $\bar{\boldsymbol{x}} = \boldsymbol{x} = \boldsymbol{x}^*$, and its dynamics given by

$$\begin{aligned}\dot{\bar{\boldsymbol{x}}} =& \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}^* \\ =& f(\boldsymbol{x}, \boldsymbol{u}) - f(\boldsymbol{x}^*, \boldsymbol{u}^*) \\ =& f(\boldsymbol{x}^* + \bar{\boldsymbol{x}}, \boldsymbol{u}^* + \bar{\boldsymbol{u}}) - f(\boldsymbol{x}^*, \boldsymbol{u}^*)\end{aligned} \tag{4.16}$$

Approximating this by the Taylor expansion around the trim point to the first order gives

$$\begin{aligned}\dot{\bar{\boldsymbol{x}}} =& f(\boldsymbol{x}^*, \boldsymbol{u}^*) + \frac{\partial f(\boldsymbol{x}^*, \boldsymbol{u}^*)}{\partial \boldsymbol{x}^T}\bar{\boldsymbol{x}} + \frac{\partial f(\boldsymbol{x}^*, \boldsymbol{u}^*)}{\partial \boldsymbol{u}^T}\bar{\boldsymbol{u}} + \text{H.O.T.} - f(\boldsymbol{x}^*, \boldsymbol{u}^*) \\ \approx& \frac{\partial f(\boldsymbol{x}^*, \boldsymbol{u}^*)}{\partial \boldsymbol{x}^T}\bar{\boldsymbol{x}} + \frac{\partial f(\boldsymbol{x}^*, \boldsymbol{u}^*)}{\partial \boldsymbol{u}^T}\bar{\boldsymbol{u}}\end{aligned} \tag{4.17}$$

Thus a linearised model for the fixed-wing UAV can be obtained by computing the Jacobians $\frac{\partial f}{\partial \boldsymbol{x}^T}$ and $\frac{\partial f}{\partial \boldsymbol{u}^T}$. Calculating the Jacobians for the longitudinal dynamics expressed in Equation (4.14) yields a state-space system given by

$$\begin{aligned}\begin{bmatrix}\dot{\bar{p}}_d \\ \dot{\bar{u}} \\ \dot{\bar{w}} \\ \dot{\bar{\theta}} \\ \dot{\bar{q}}\end{bmatrix} =& \begin{bmatrix}0 & -\sin\theta^* & \cos\theta^* & -u^*\cos\theta^* - w^*\sin\theta^* & 0 \\ 0 & X_u & X_w & -g\cos\theta^* & X_q \\ 0 & Z_u & Z_w & -g\sin\theta^* & Z_q \\ 0 & 0 & 0 & 0 & 1 \\ 0 & M_u & M_w & 0 & M_q\end{bmatrix}\begin{bmatrix}\bar{p}_d \\ \bar{u} \\ \bar{w} \\ \bar{\theta} \\ \bar{q}\end{bmatrix} \\ &+ \begin{bmatrix}0 & 0 \\ X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & 0 \\ 0 & 0 \\ M_{\delta_e} & 0\end{bmatrix}\begin{bmatrix}\bar{\delta}_e \\ \bar{\delta}_t\end{bmatrix}\end{aligned} \tag{4.18}$$

The coefficients are given in Table (4.2). The thrust and pitch from the quadcopter part has not been included here.

X_u	$\frac{u^*\rho S}{m}(C_X(\alpha^*) + C_{Z_{\delta_e}}\delta_e^*) - \frac{w^*\rho S C_{X_\alpha}}{2m}$
	$+\frac{u^* q^* \rho S c C_{X_q}}{4mV_a^*} - \frac{u^*\rho S_{prop} C_{prop}}{m}$
X_w	$-q^*\frac{w^*\rho S}{m}(C_X(\alpha^*) + C_{X_{\delta_e}}\delta_e^*) + \frac{w^*\rho S C_{X_\alpha}}{2m}$
	$+\frac{w^* q^* \rho S c C_{X_q}}{4mV_a^*} - \frac{w^*\rho S_{prop} C_{prop}}{m}$
X_q	$-w^* + \frac{\rho S c V_a^* C_{X_q}}{4m}$
X_{δ_e}	$\frac{\rho S V_a^{*2} C_{X_{\delta_e}}}{2m}$
X_{δ_t}	$\frac{\rho S_{prop} C_{prop} k_{motor}^2 \delta_t^*}{m}$
Z_u	$q^* + \frac{u^*\rho S}{m}(C_Z(\alpha) + C_{Z_{\delta_e}}\delta_e^*) - \frac{w^*\rho S C_{Z_\alpha}}{2m}$
	$\frac{u^* q^* \rho S c C_{Z_q}}{4mV_a^*}$
Z_w	$\frac{w^*\rho S}{m}(C_Z(\alpha) + C_{Z_{\delta_e}}\delta_e^*) + \frac{u^*\rho S C_{Z_\alpha}}{2m}$
	$\frac{w^* q^* \rho S c C_{Z_q}}{4mV_a^*}$
Z_q	$u^* + \frac{\rho S c V_a^* C_{Z_q}}{4m}$
Z_{δ_e}	$\frac{\rho S V_a^{*2} C_{Z_{\delta_e}}}{2m}$
M_u	$\frac{u^*\rho S c}{J_{yy}}(C_{M_0} + C_{M_\alpha} + C_{M_{\delta_e}}\delta_e^*) - \frac{w^*\rho S c C_{M_\alpha}}{2J_{yy}}$
	$\frac{u^*\rho S c^2 C_{M_q}}{4J_{yy}V_a^*}$
M_q	$\frac{\rho S c^2 V_a^* C_{M_q}}{4J_{yy}}$
M_{δ_e}	$\frac{\rho S c V_a^{*2} C_{M_{\delta_e}}}{2J_{yy}}$

Table 4.2: Coefficients for the linearised state-space system of the longitudinal fixed-wing UAV dynamics. $C_X(\alpha^*)$, $C_Z(\alpha^*)$ are given in Equation(3.43)

4.3.3.3 Non-Linear Observer Design

Under the assumption of small angles of attack, and considering lateral movements as negligible Equation (4.14) describe the three degrees of freedom

non-linear longitudinal dynamics of a fixed-wing UAV. Following the approach of [39] Equation (4.14) can be brought to a continuous time quasi-LPV form by embedding the non-linearities into the parameters in the following way:

$$\dot{\boldsymbol{x}} = A(u, w, \theta, q)\boldsymbol{x} + B(u, w)\boldsymbol{u} + d(\theta) \tag{4.19}$$

with the state given by $\boldsymbol{x} = [p_d, u, w, \theta, q]^T$ and control input $\boldsymbol{u} = [\delta_e, \delta_t]^T$. The matrices A, B and d are given by

$$\begin{aligned} A(u, w, \theta, q) &= \begin{bmatrix} 0 & a_{12} & a_{13} & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 & a_{25} \\ 0 & a_{32} & a_{33} & 0 & a_{35} \\ 0 & 0 & 0 & 0 & 1 \\ 0 & a_{52} & a_{53} & 0 & a_{55} \end{bmatrix} \\ B(u, w) &= \begin{bmatrix} 0 & 0 \\ b_{21} & b_{22} \\ b_{31} & 0 \\ 0 & 0 \\ b_{51} & 0 \end{bmatrix}, \qquad d(\theta) = \begin{bmatrix} 0 \\ -g\sin\theta \\ g\cos\theta \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{4.20}$$

where the matrix parameters are listed in Table (4.3)

a_{12}	$-\sin\theta$
a_{13}	$\cos\theta$
a_{22}	$\frac{u\rho S}{2m}(C_X(\alpha) - \frac{S_{prop}C_{prop}}{S})$
a_{23}	$\frac{w\rho S}{2m}(C_X(\alpha) - \frac{S_{prop}C_{prop}}{S})$
a_{25}	$-w + \frac{\rho S c V_a}{4m}C_{X_q})(\alpha)$
a_{32}	$\frac{u\rho S}{2m}(C_Z(\alpha))$
a_{33}	$\frac{w\rho S}{2m}(C_Z(\alpha))$
a_{35}	$u + \frac{\rho S c V_a}{4m}C_{Z_q})(\alpha)$
a_{52}	$\frac{u\rho S c}{2J_{yy}}(C_{M_0} + C_{M_\alpha})$
a_{53}	$\frac{w\rho S c}{2J_{yy}}(C_{M_0} + C_{M_\alpha})$
a_{55}	$\frac{\rho S c^2 V_a}{4J_{yy}}$
b_{21}	$\frac{\rho S V_a^2}{2m}C_{X_{\delta_e}}$
b_{22}	$\frac{\rho S_{prop}C_{prop}}{2m}k^2_{motor}$
b_{31}	$\frac{\rho S V_a^2}{2m}C_{Z_{\delta_e}}$
b_{51}	$\frac{\rho S c V_a^2}{2J_{yy}}C_{M_{\delta_e}}$

Table 4.3: Coefficients for the state-space system of the longitudinal fixed-wing UAV dynamics with non-linearities embedded in the system matrices. $C_X(\alpha^*)$, $C_Z(\alpha^*)$ are given in Equation (3.43)

4.3.4 Longitudinal Autopilot Design

Utilising the linearised state-space system described in Equation (4.18) it is possible to design an autopilot for the longitudinal system in order drive the aircraft to a desired altitude p_d^d and airspeed V_a^d using LQR control with integral action [45][1]. Following the approach in Section (3.4.8) to construct a system to control attitude and airspeed, the integral state is given by

[1]Web-page by authors of [35], the controller is given in a currently unpublished supplement to the book

$$\begin{aligned} \boldsymbol{x}_I &= \begin{bmatrix} \int (p_d - p_d^d)\, dt \\ \int (V_a - V_a^d)\, dt \end{bmatrix} \\ &= \int \left[H_{lon}\boldsymbol{x} - \begin{bmatrix} p_d^d \\ V_a^d \end{bmatrix} \right], \end{aligned} \tag{4.21}$$

where

$$H_{lon} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{u^*}{V_a^*} & \frac{w^*}{V_a^*} & 0 & 0 \end{bmatrix}. \tag{4.22}$$

It is then possible to express the state-space matrices for the augmented system by

$$\bar{A} = \begin{bmatrix} A & 0 \\ H_{lon} & 0 \end{bmatrix}, \qquad \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \tag{4.23}$$

which can then be used for computing the LQR feedback gain in the usual manner. The specific weighting matrices used were

$$Q_{LQR} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.24}$$

$$R_{LQR} = \begin{bmatrix} 1000 & 0 \\ 0 & 10000 \end{bmatrix}$$

For clarification, the specific control input $\boldsymbol{u}_t^i$ used is given by

$$\boldsymbol{u}_t^i = \boldsymbol{u}^{*i} - K\left(\begin{bmatrix} \hat{\boldsymbol{x}}_t^i \\ \boldsymbol{x}_{I,t}^i \end{bmatrix} - \begin{bmatrix} \boldsymbol{x}^{*i} \\ \mathbf{0}_{2\times 1} \end{bmatrix} \right) + \begin{bmatrix} 0 \\ 0.01 \sin 8t \end{bmatrix}. \tag{4.25}$$

$\boldsymbol{u}^{*i}$ and $\boldsymbol{x}^{*i}$ are the trim conditions for model i with the first term in $\boldsymbol{x}^{*i}$ replaced with the desired altitude p_d^d. $\hat{\boldsymbol{x}}_t^i$ is the model-specific estimate from Equation (3.75). The sine wave is added to ensure persistent excitation.

Chapter 5

Results

5.1 Single Axis Simulation

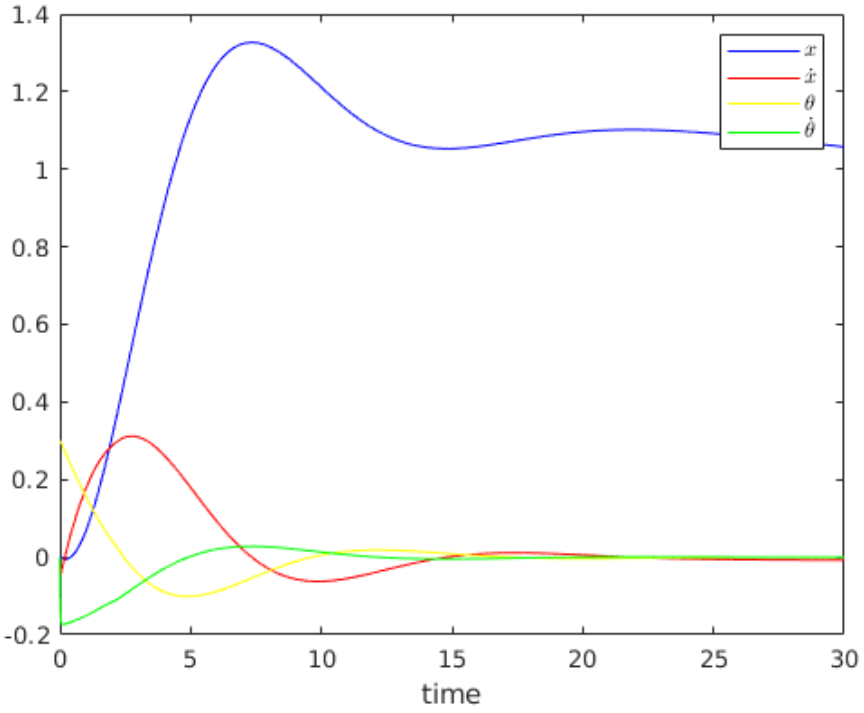

Figure 5.1: The system states of a simulation of Equation (4.1) using the multiple model adaptive estimation and control algorithm.

Figure (5.1) shows the estimated state values for a simulation based on the tailsitter model. The controller is able to drive the system towards the desired setpoint. In addition, the MMA algorithm is able to correctly identify the candidate model with $\epsilon = 0.525$ which is the one closest to the real value of $\epsilon =$

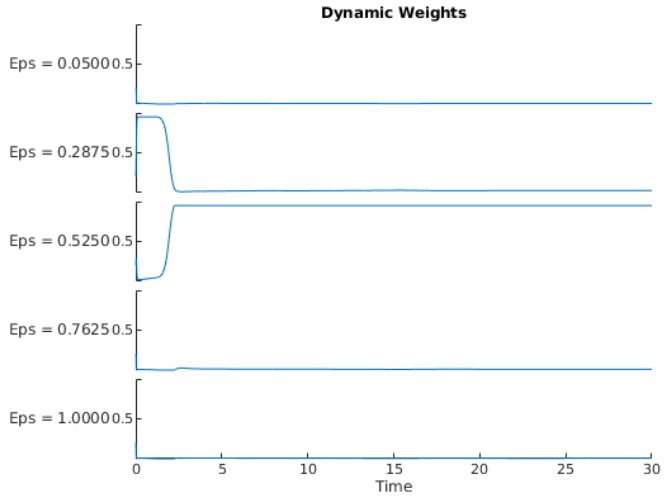

Figure 5.2: The dynamic weights for each of the candidate models from simulation of the model in Equation (4.1). The labels on the left are the value of the unknown parameter ϵ in each of the candidate models. The true parameter value in the plant was $\epsilon = 0.45$

0.45, even if it initially decided on the wrong model. The dynamic weighting of the different candidate models is shown in Figure (5.2).

5.2 Quadcopter 6DoF Non-linear Plant Simulation

The non-linear dynamic model given by Equations (3.38-3.41) with external forces as described in Section (4.2.1) was used as the system plant. The quadcopter was modelled as a $1m \times 1m \times 0.2m$ box. The non-symmetry in the mass and inertia was modelled as a point mass at position $[x_b, y_b, z_b]^T = [-1.5, 0.2, 0]^T$ relative to the CM of the quadcopter box. For the different candidate models for the multiple-model algorithm the value of the point mass was set as the unknown parameter. For each of the simulations five candidate models were set up in the range $m_{point} \in [0.5, 5]$. A few different approaches for simulations were investigated in order to analyse the behaviour of the multiple-model algorithm as well as the response of the controller.

5.2.1 Hovering

In the first simulation the quadcopter initial position was $\boldsymbol{x}_0 = \mathbf{0}_{12\times1}$. It was set to hover at two different heights. The the inertial positions can be seen in Figure (5.3), the Euler-angle positions can be seen in Figure (5.4). It might seem overly redundant to include all 6 positions when the only visible actions happen in the altitude. This is to keep consistency with the figures that will

appear in the following sections as well as showing that the quadcopter is stable for all degrees of freedom and stays within the limits of the linearisation. The true value of the parameter in the plant was set to $m_{point} = 2.0$, and Figure (5.5) shows the dynamic weights of each of the 5 candidate models for the simulation. The MMA algorithm instantly choose the candidate model with $m_{point} = 1.6$, which is the candidate model with the parameter value most similar to the plant parameter value.

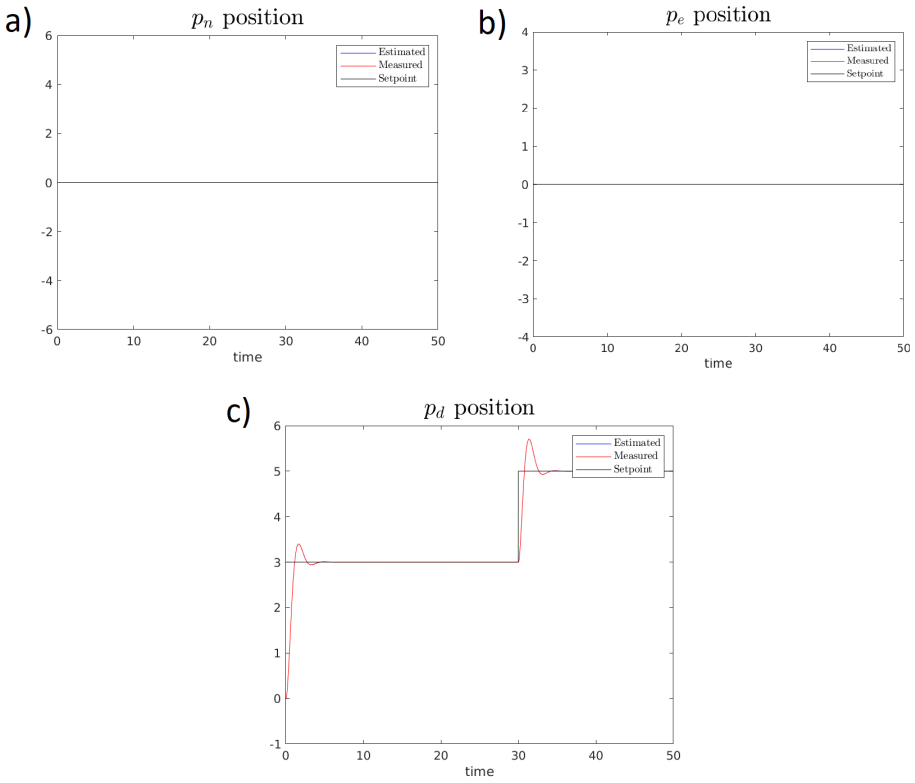

Figure 5.3: Inertial positions from simulation when quadcopter was set to hover. a) p_n-position, b) p_e-position and c) altitude $-p_d$

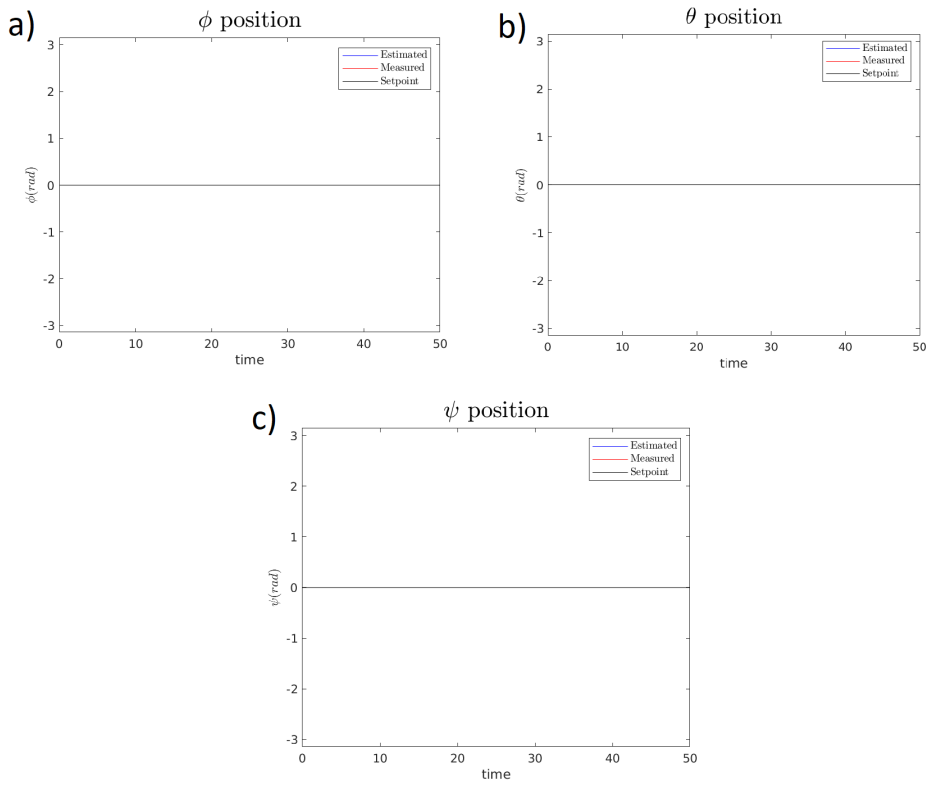

Figure 5.4: The Euler-angle positions when the quadcopter was set to hover. a) ϕ-angle, b) θ-angle and c) the ψ-angle

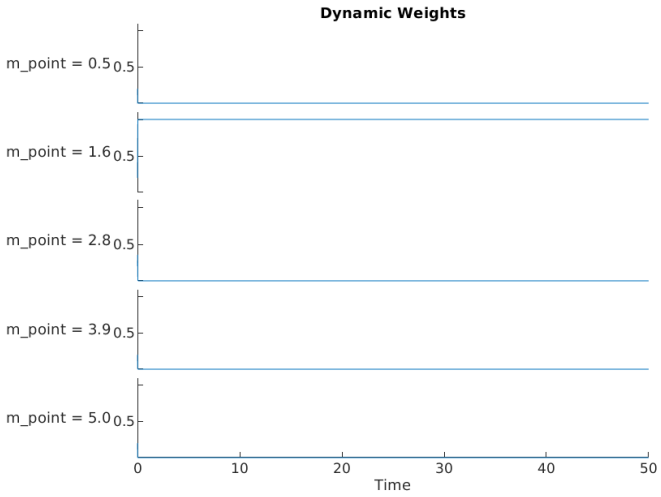

Figure 5.5: Dynamic weighting of each of the candidate models, with the parameter value of each model indicated by the m_{point}-value to the left of each plot.

5.2.2 Step Changes in Setpoint

Figures (5.6) and (5.8) show the inertial and Euler-angle positions from a simulation the quadcopter was set to track varying step changes in desired setpoint values in all inertial directions. The quadcopter was set to initial position $\boldsymbol{x}_0 = [3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$. For this simulation the plant value of the $m_{point} = 2.0$ was used. Figure (5.7) shows the dynamic weights for each of the candidate models with the parameter value of the respective models given on the left in the figure.

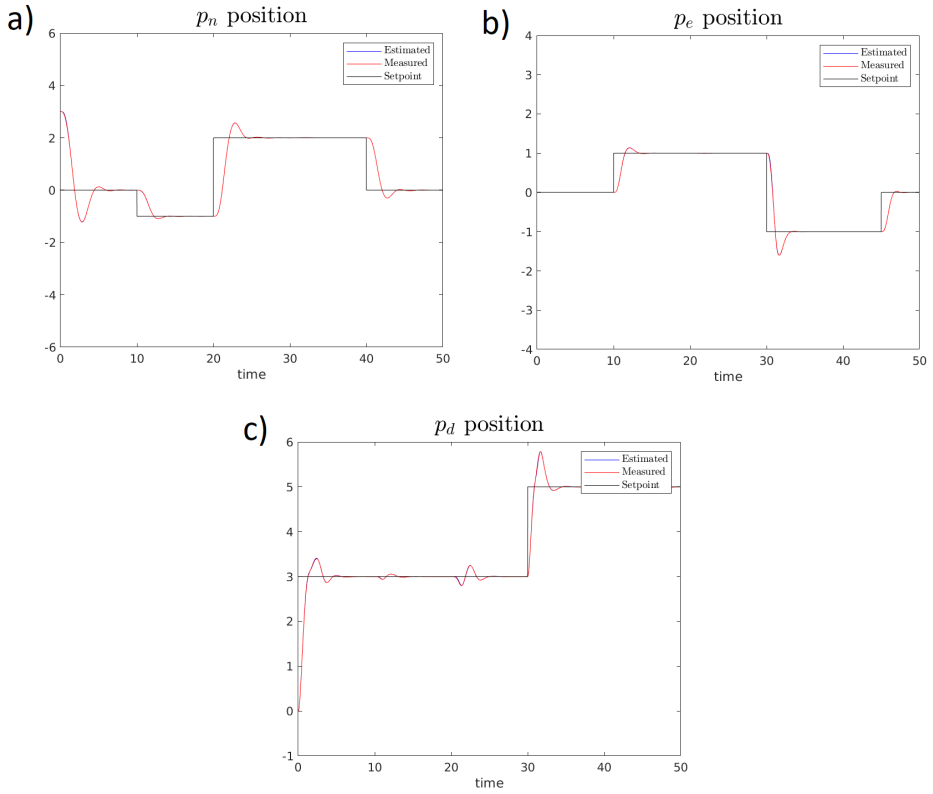

Figure 5.6: Inertial positions from a simulation when quadcopter was set to track step changes in desired setpoint values. a) p_n-position, b) p_e-position and c) altitude $-p_d$

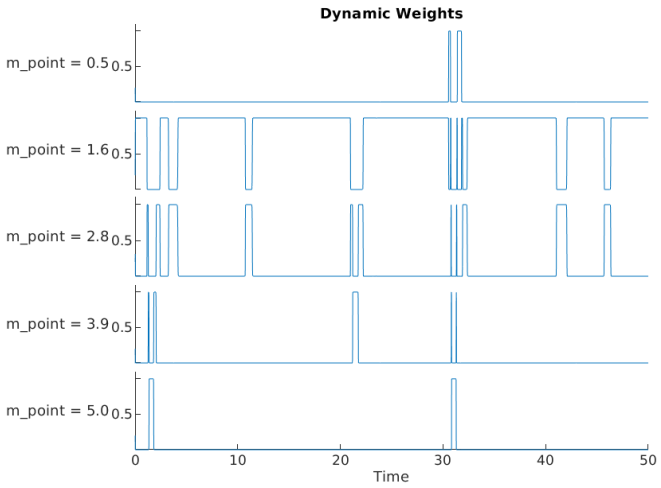

Figure 5.7: Dynamic weighting of the candidate models when the quadcopter tracked step changes in setpoint values. The parameter value of each model indicated by the m_{point}-value to the left of each plot.

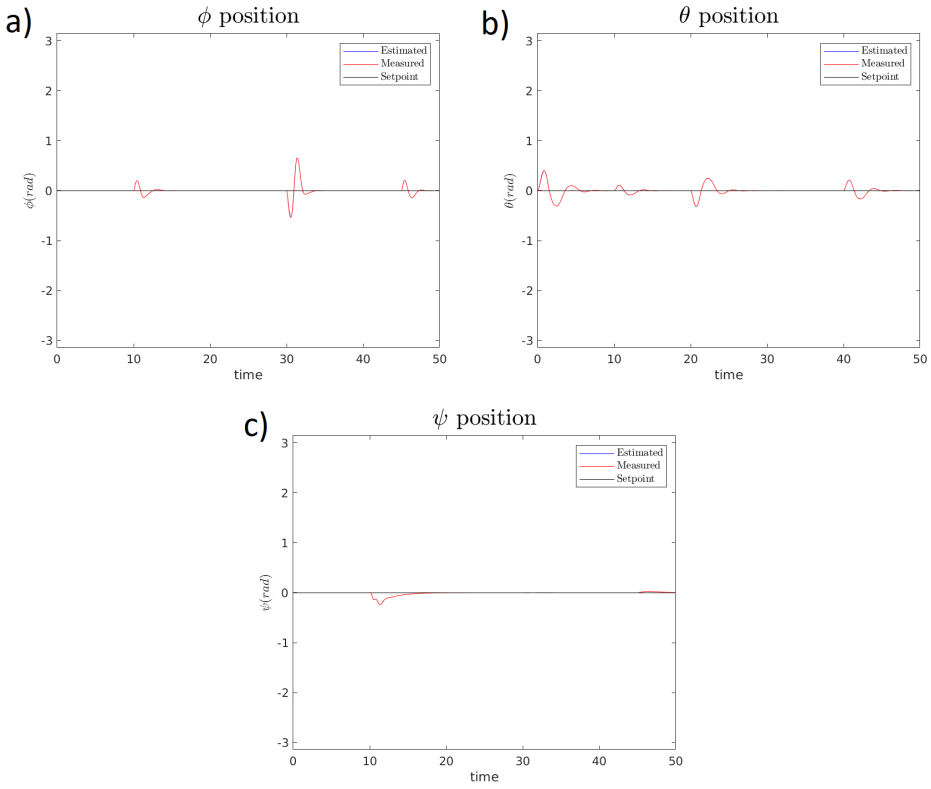

Figure 5.8: The Euler-angle positions when the quadcopter was set to track step changes in desired setpoint values. a) shows ϕ-angle, b) θ-angle and c) the ψ-angle

5.2.3 Trajectory Tracking

Figures (5.9) and (5.10) show the inertial and angular positions of the quadcopter set to follow a circular trajectory in the $p_n p_e$-plane as well as two step changes in desired altitude. The initial position was set to $\boldsymbol{x}_0 = [3,0,0,0,0,0,0,0,0,0,0,0]^T$, the plant parameter was the same $m_{point} = 2$ and five candidate models were used. Figure (5.11) shows the dynamic weights for each of the candidate models with the parameter value of the respective models given on the left in the figure. Figure (5.10) shows the Euler-angle positions for the simulation.

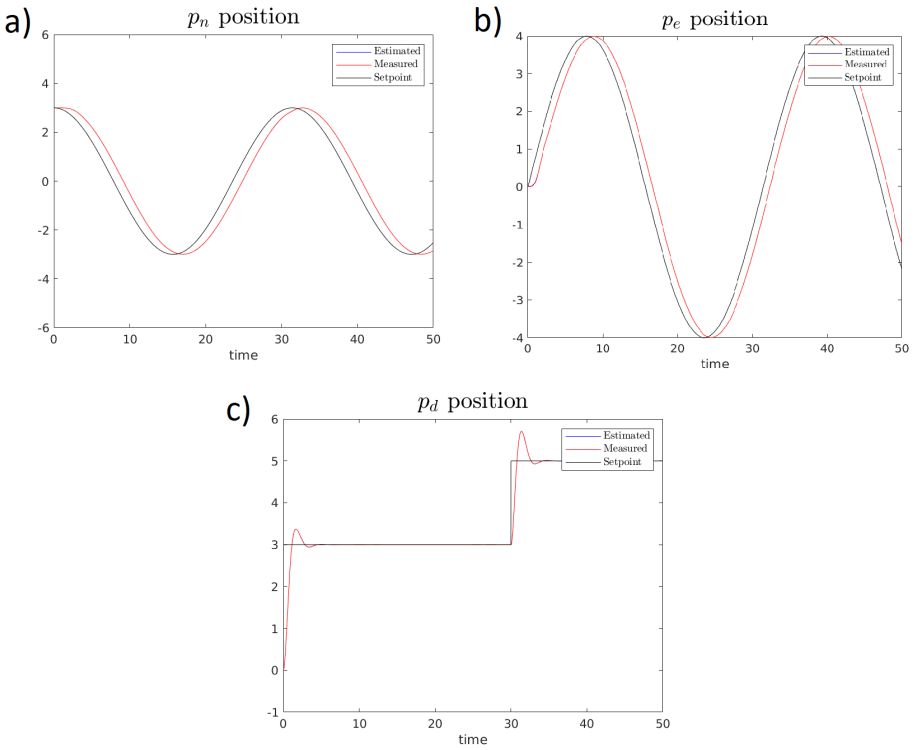

Figure 5.9: Inertial positions when the quadcopter was set to track a trajectory in $p_n p_e$-direction as well as step changes in height $-p_d$. a) p_n-position, b) p_e-position and c) altitude $-p_d$

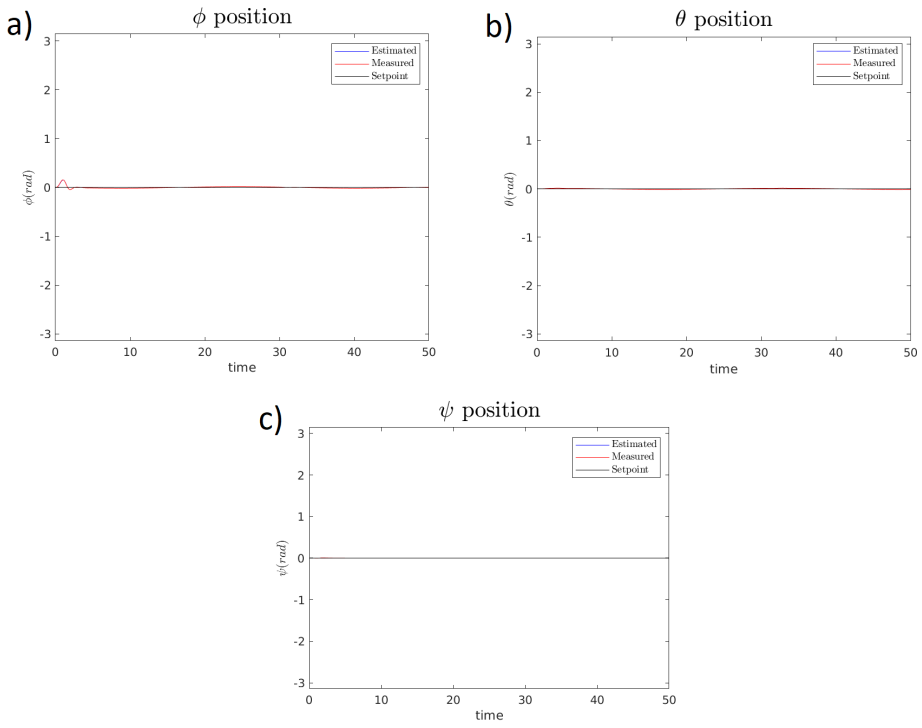

Figure 5.10: Euler angle positions when the quadcopter was set to track a trajectory in $p_n p_e$-direction as well as step changes in height h. a) shows ϕ-angle, b) θ-angle and c) the ψ-angle

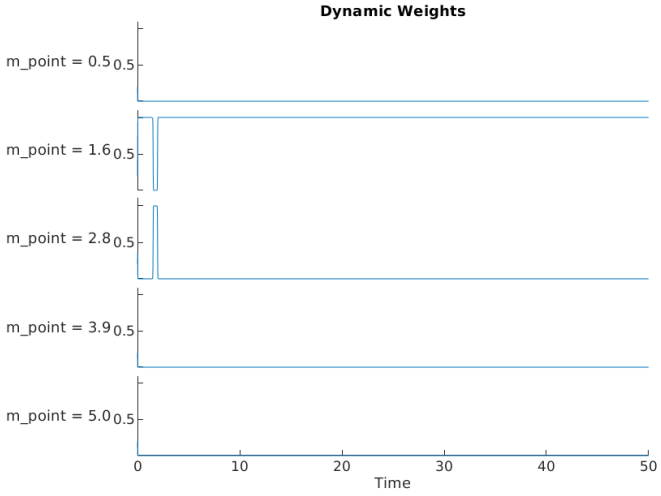

Figure 5.11: The dynamic weighting of each of the candidate models when the quadcopter was set to to track a trajectory in $p_n p_e$-direction as well as step changes in altitude setpoint. The parameter value of each model indicated by the m_{point}-value to the left of each plot. The true parameter value was $m_{point} = 2.0$

5.2.4 Variable Weight

In this section the quadcopter was set to follow the same trajectory as in Section(5.2.3) with the same initial position x_0. But, instead of having the plant parameter m_{point} set to a constant value it was instead changed at regular intervals. Figure (5.15) illustrate how the value of the parameter in the plant started off at $m_{point} = 0.1$, well below the lowest value of the candidate models, and increase by 1 every 10 seconds. The dashed lines in the figure indicate the parameter value for each of the candidate models. Figure (5.14) shows the dynamic weight of each of the candidate models.

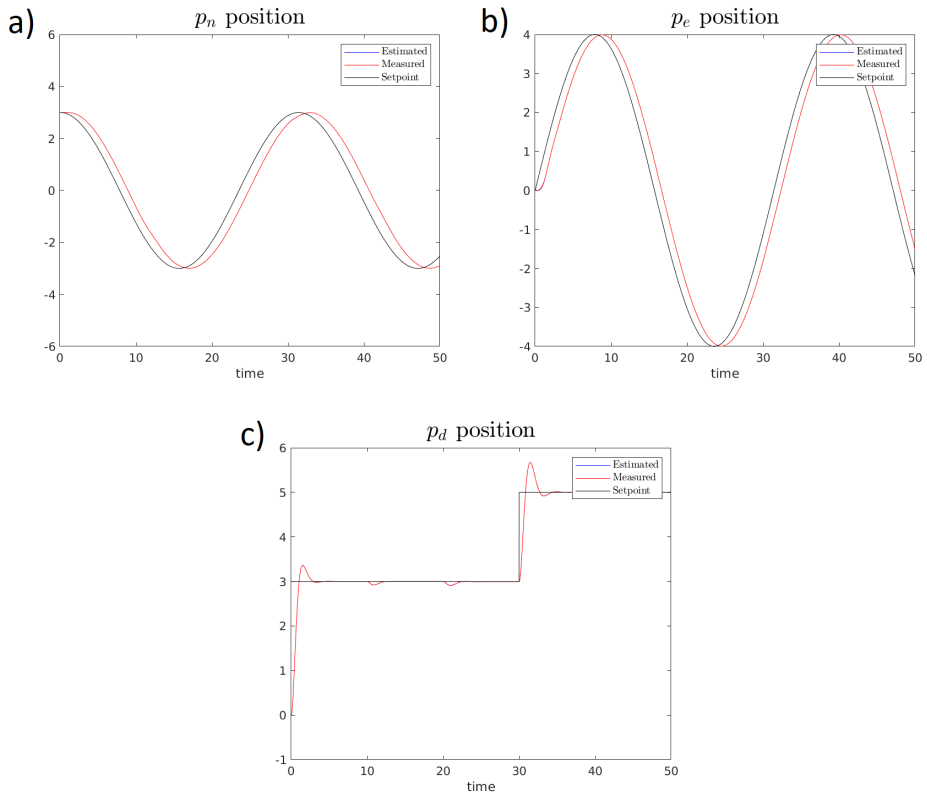

Figure 5.12: Inertial positions when the quadcopter was set to track a trajectory in the $p_n p_e$-directions as well as step changes in altitude $-p_d$ while the plant parameter value m_{point} changed. a) shows p_n-position, b) p_e-position and c) altitude $-p_d$

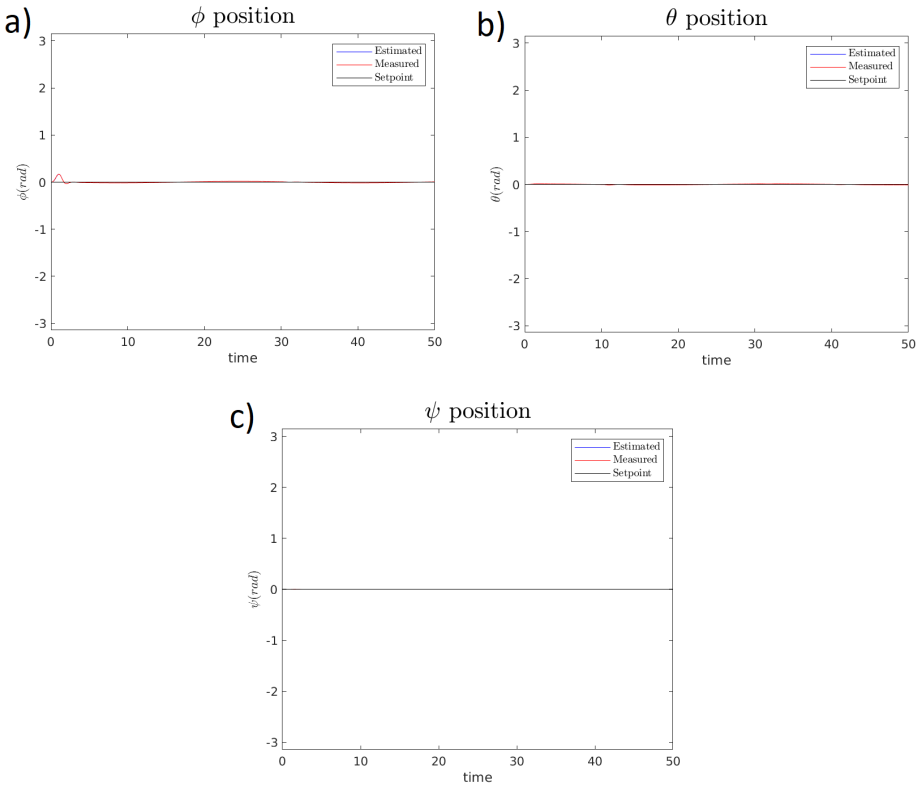

Figure 5.13: Euler angle positions when the quadcopter was set to track a trajectory in the $p_n p_e$-directions as well as step changes in altitude while the plant parameter value m_{point} changed. a) ϕ-angle, b) θ-angle and c) ψ-angle

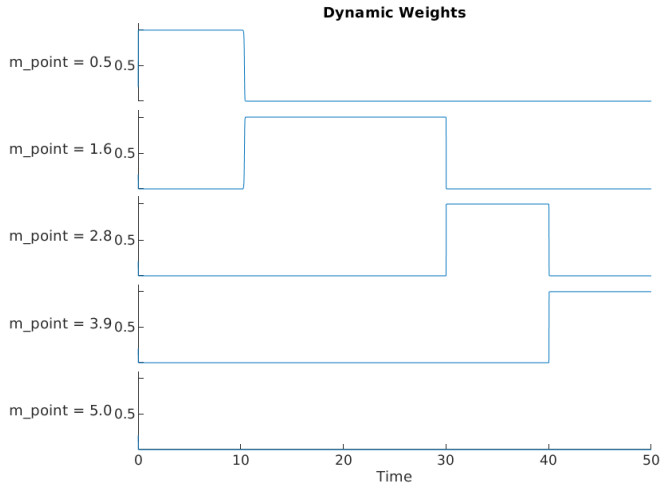

Figure 5.14: Dynamics weighting for each of the candidate models when the quadcopter tracked trajectory while the plant parameter value m_{point} changed. The parameter value of each model is indicated by the m_{point}-value to the left of each plot. The true parameter value was initially set to $m_{point} = 0.1$, and incremented with 1 every 10 seconds, as illustrated in in Figure (5.15)

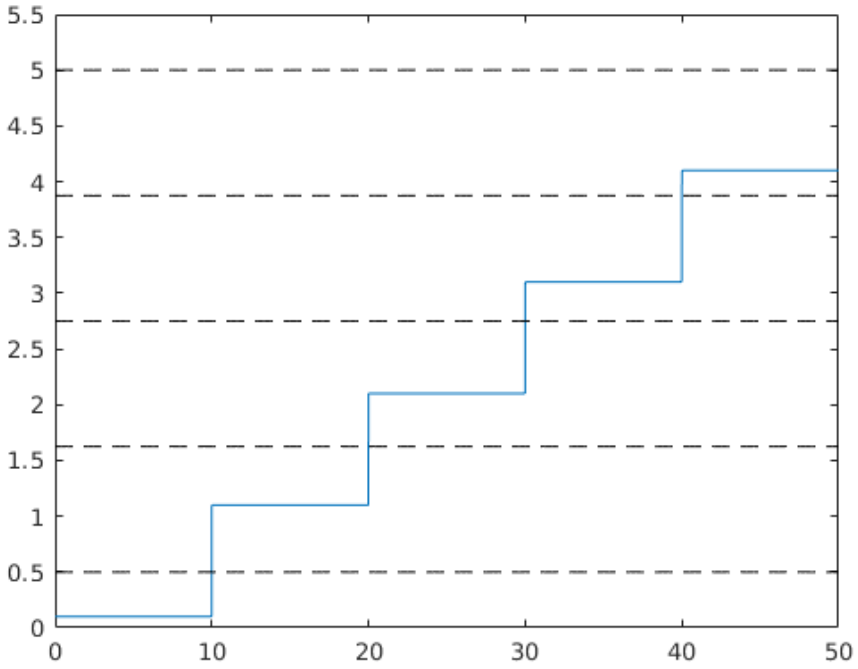

Figure 5.15: Plot illustrating how the plant parameter value m_{point} changed throughout the simulation. Blue line shows the parameter value. The black dotted lines show the parameter values for each of the candidate models.

5.3 Fixed-Wing Longitudinal Flight Simulation

The longitudinal simulation model for the fixed-wing aircraft was implemented as described in Section (4.3). A program was implemented in *MATLAB* and *Simulink* with inspiration from [35]. The user could insert desired airspeed V_a and desired flight path angle γ for the aircraft. The program would find the corresponding trim conditions $\boldsymbol{x}^* = [-, u^*, w^*, \theta^*, q^*]^T$ and $\boldsymbol{u} = [\delta_e^*, \delta_t^*]^T$ for that specific airspeed and flight angle. The $-$ in the trim state means that the p_d position was assumed a free variable when computing the trim conditions since the altitude is irrelevant for maintaining stable flight, under the assumption of altitude ranges of constant air density. Computation of the trim conditions for the longitudinal state was done while only considering the longitudinal dynamics from Section (4.3.1). This did not affect the longitudinal state trim conditions considerably and was done under the assumption that the lateral dynamics could be considered a small disturbance.

For a given set of trim conditions a state estimator was designed by embedding the non-linearities as described in Section (4.3.3.3). With the same trim conditions a corresponding LQR controller was designed by computing the linearised state-space system as given in Section (4.3.3.2). This corresponds to a single model within the MMA LPV paradigm.

The trim conditions for simulations of the fixed-wing aircraft were all computed according to a desired airspeed $V_a^* = 25m/s$ and flight path angle $\gamma^* = 0$. The initial conditions for the simulations were $\boldsymbol{x}_0 = [0,\ 25,\ 0.1,\ 0.1,\ 0]^T$.

5.3.1 Single Model Simulation

The first simulation was designed in order to verify that the controller was able to maintain stable flight in accordance with the specified trim conditions, as well as follow a trajectory in altitude and airspeed. To create a reference point for future simulations. These simulations were done using only a single model with parameters corresponding to the true parameters of the aircraft, and only a single LPV model was considered. Figure (5.16) shows a plot of the estimated, measured and desired altitude ($-p_d$). Figure (5.17) shows the same for the airspeed. Figure (5.18) shows the attitude of the aircraft during level flight.

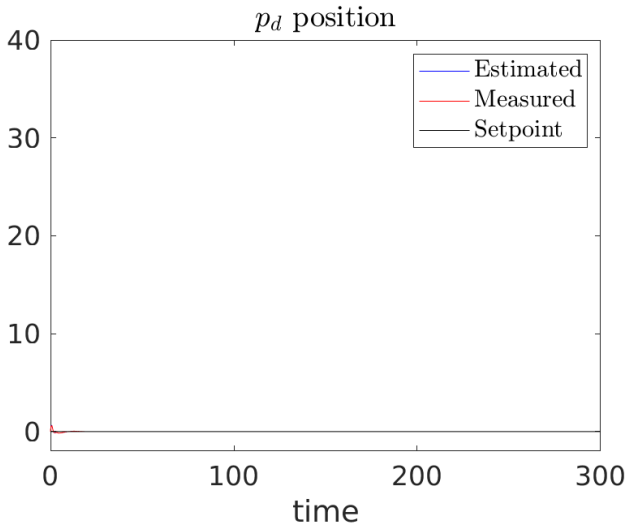

Figure 5.16: The estimated, measured and desired altitude $-p_d$ when the aircraft was set to maintain steady level flight with a single model.

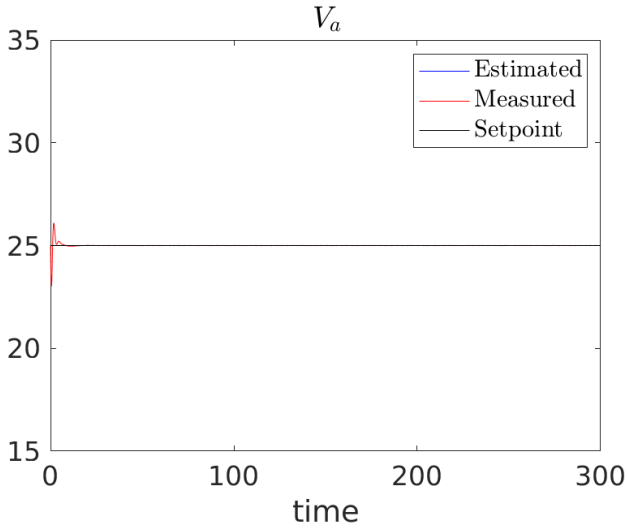

Figure 5.17: The estimated, measured and desired airspeed V_a when the aircraft was set to maintain a constant airspeed equal to the trim point airspeed with a single model.

In the next simulation with only a single model the aircraft was set to follow a desired trajectory in airspeed and altitude. Figure (5.19) shows the altitude of the aircraft together with the desired altitude trajectory, and Figure (5.20) shows the same for the airspeed trajectory. Figure (5.21) shows the estimated and measured attitude.

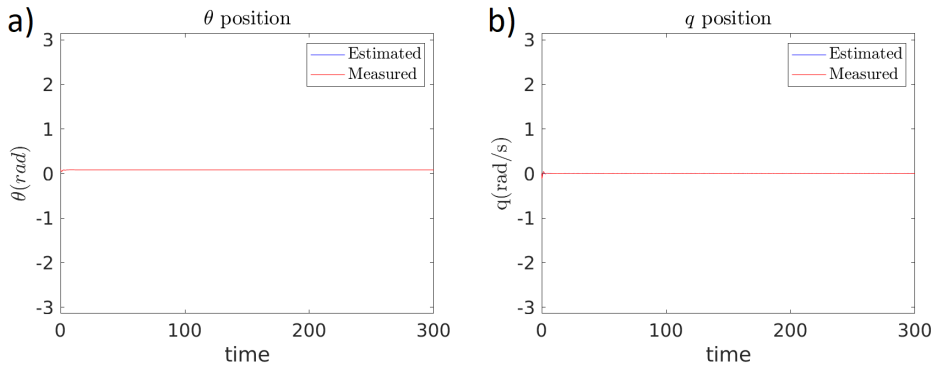

Figure 5.18: a) The measured and estimated pitch θ. b) The measured and estimated pitch rate q. Both plots are from the simulation when the aircraft was set to maintain a constant level flight.

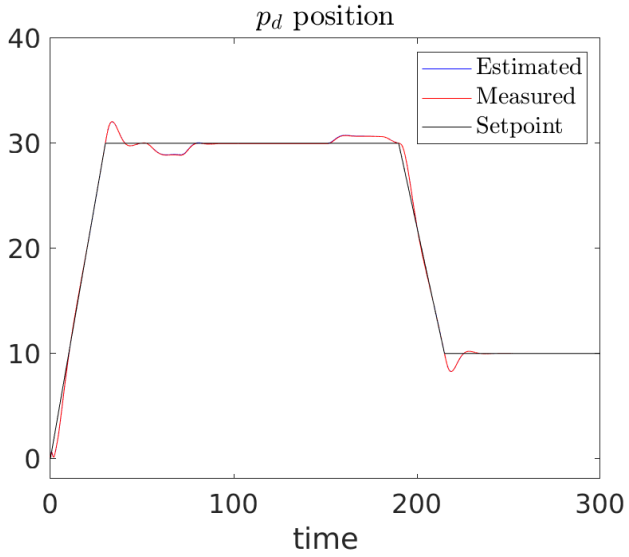

Figure 5.19: The estimated, measured and desired altitude $-p_d$ when the aircraft was set to follow a trajectory with a single model.

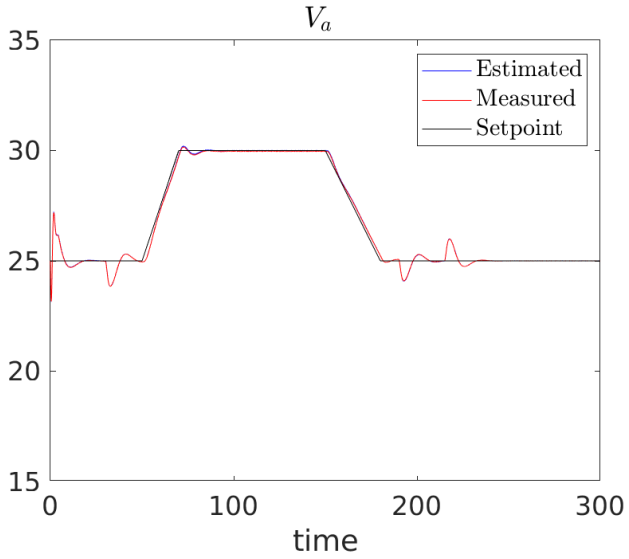

Figure 5.20: The estimated, measured and desired airspeed V_a when the aircraft was set to follow a trajectory in desired airspeed with a single model.

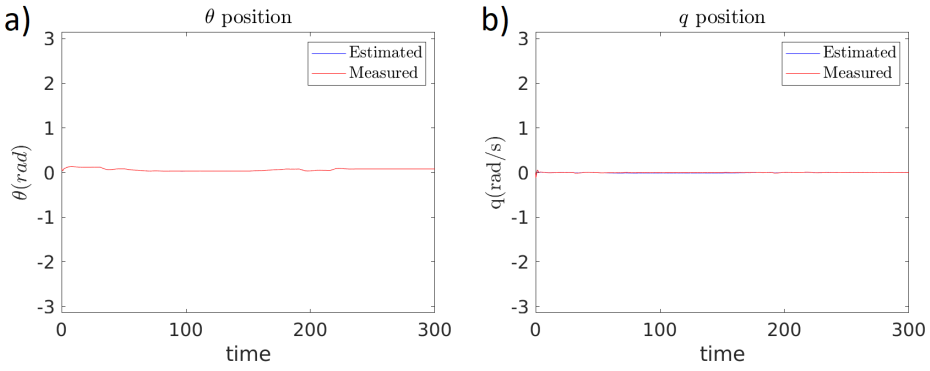

Figure 5.21: a) The measured and estimated pitch θ. b) The measured and estimated pitch rate q. The plots were obtained in a simulation when the aircraft was set to follow a trajectory in desired altitude and desired airspeed with a single model.

5.3.2 Fixed-wing Multiple Model Adaptive Estimation and Control

The next step in the simulations was to implement the multiple model adaptive estimation and control algorithm. In these simulations it was assumed that the aerodynamic coefficient C_{L_α} in Equation (4.15) was unknown. With the true value of $C_{L_\alpha} = 3.45$, five candidate models were constructed with parameter values uniformly distributed within the range $C_{L_\alpha} \in [2.55, 4.55]$. Estimators and controllers were constructed for each candidate model in the same manner as for the single model simulations and the trim points were all calculated according to airspeed $V_a^* = 25m/s$ and flight path angle $\gamma^* = 0$. The initial conditions were $\boldsymbol{x}_0 = [0, 25, 0.1, 0.1, 0]^T$. Figure (5.22) shows the altitude for a simulation where the aircraft was set to maintain level flight, and Figure (5.23) the airspeed. Figure (5.24) shows the pitch angle and pitch rate. The true value of $C_{L_\alpha} = 3.45$ was used in the plant, Figure (5.25) shows the dynamic weight for each of the candidate models with the parameter value of each model given on the left in the figure.

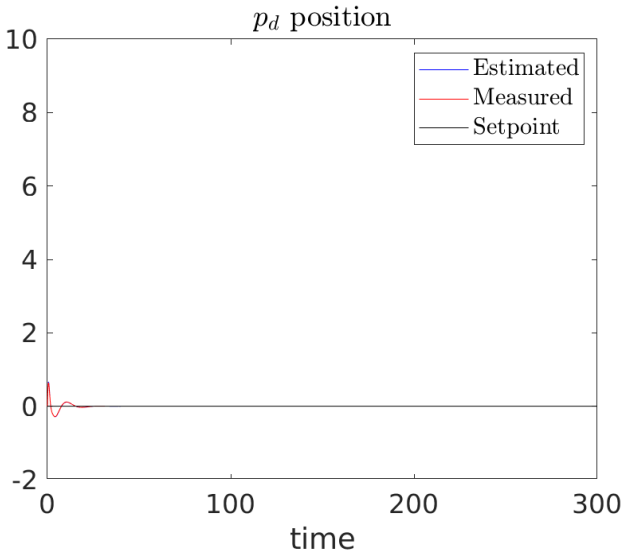

Figure 5.22: The estimated, measured and desired altitude $-p_d$ when the aircraft was set to maintain steady level flight using the MMA algorithm with five candidate models.

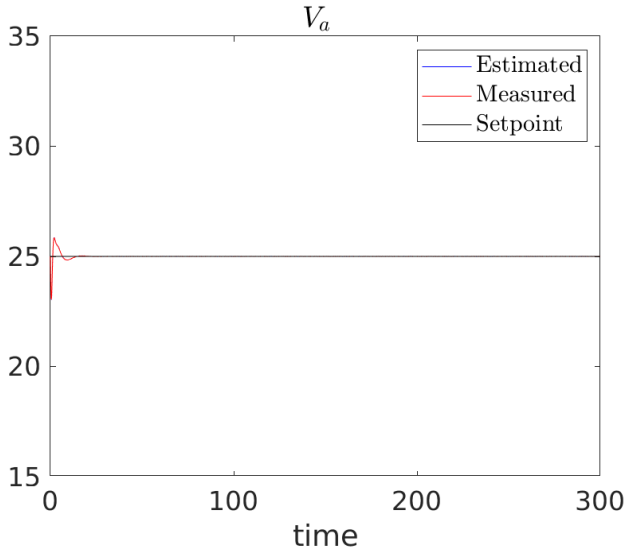

Figure 5.23: The estimated, measured and desired airspeed V_a when the aircraft was set to maintain a constant airspeed equal to the trim point airspeed. The MMA algorithm was used with five candidate models.

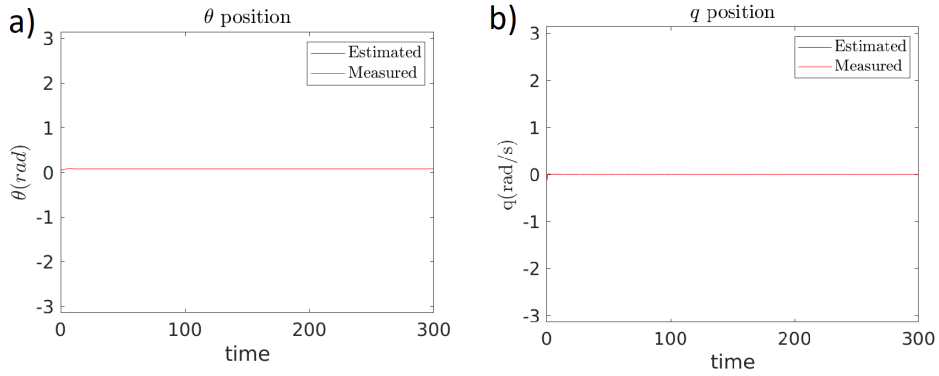

Figure 5.24: a) The estimated and measured pitch θ. b) The estimated and measured pitch rate q. Both plots were obtained for the simulation were the aircraft was set to maintain constant level flight while using five candidate models to estimate and control the system.

In the next simulations the aircraft was set to follow the same trajectory in airspeed and altitude as the single model simulation, but now with the value of C_{L_α} in the plant assumed unknown. Figures (5.26) and (5.27) show the estimated, measured and desired altitude and airspeed respectively. Figure (5.28) shows the pitch angle and pitch rate. Figure (5.30) shows a zoomed-in plot of the pitch angle which will be discussed in the next chapter. Figure (5.29) shows the dynamic weights of the candidate models.

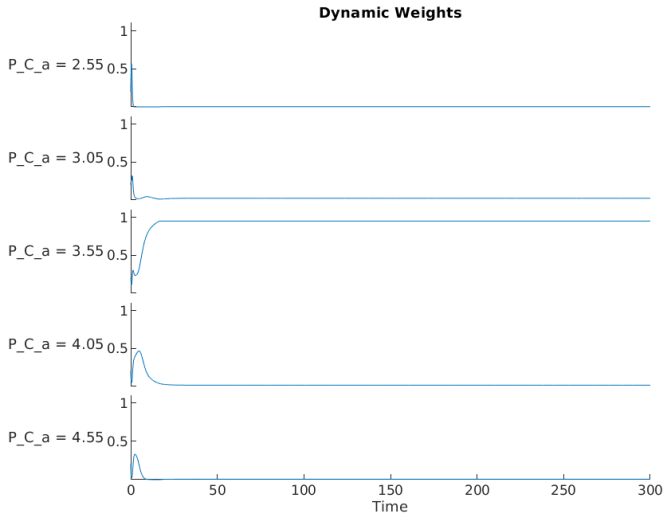

Figure 5.25: Dynamic weight for the simulation where the aircraft was set to maintain level flight. Label on the left indicates the C_{L_α} value for the corresponding candidate model. A true value of $C_{L_\alpha} = 3.45$ was used for the plant.

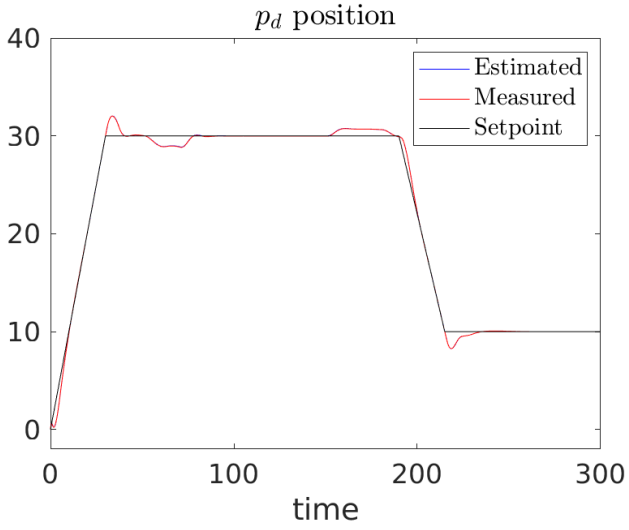

Figure 5.26: The estimated, measured and desired altitude $-p_d$ when the aircraft was set to follow a trajectory using multiple model adaptive estimation and control with five candidate models

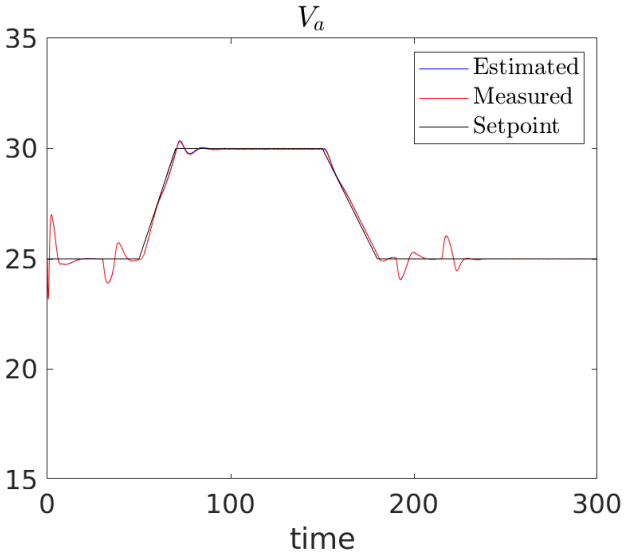

Figure 5.27: The estimated, measured and desired airspeed V_a when the aircraft was set to follow a trajectory in desired airspeed. This was done while using five candidate models.

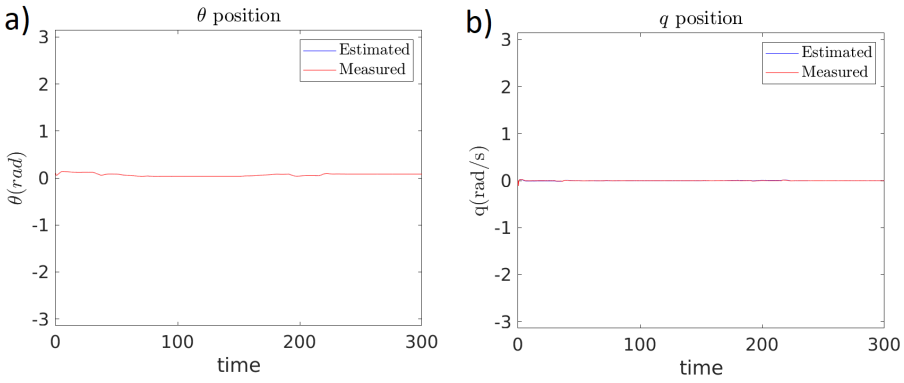

Figure 5.28: a) The estimated and measured pitch θ. b) The estimated and measured pitch rate q. Both plots were obtained when the aircraft was set to follow a trajectory in desired altitude and desired airspeed while using five candidate models to estimate and control the system.

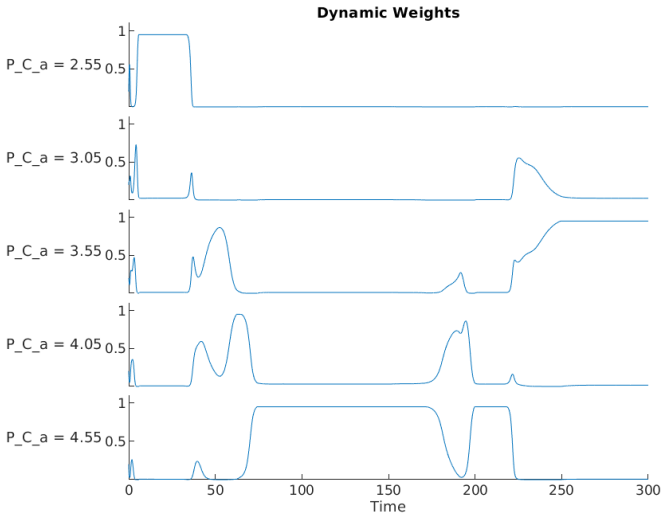

Figure 5.29: Dynamic weights for the candidate models when the aircraft was set to follow a trajectory of desired airspeed and desired altitude. Label on the left indicates the C_{L_α} value for the corresponding candidate model. A true value of $C_{L_\alpha} = 3.45$ was used for the plant.

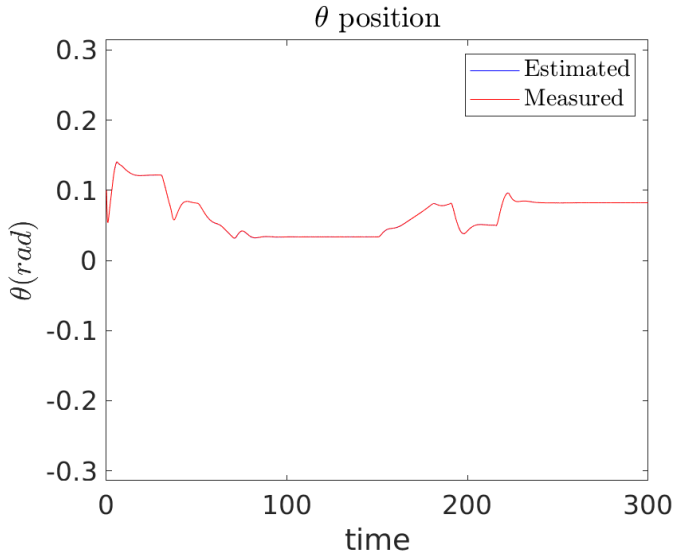

Figure 5.30: A zoomed-in version of Figure (5.28 a).

5.3.3 Fixed-wing LPV Control

Linear parameter varying methods are expected to be a good solution to the problems that occurred when the aircraft followed trajectories in airspeed and altitude. However, no positive results were acquired. Following will be a description on the approach that was attempted using only LPV estimation and control without considering multiple models. The parameters of the estimation model was set to be equal to the plant.

Seeing that the variations in airspeed and angle of attack caused problems for the MMA algorithm without the LPV methods it was chosen to use the airspeed $V_a = \sqrt{u^2 + w^2}$ and the flight path angle $\gamma = \arcsin \dot{h}/Va$, with $\dot{h} = u \sin\theta - w \cos\theta$, as the scheduling parameters $\boldsymbol{v}(t)$. These were conveniently the same parameters used as inputs to the program that calculated the trim conditions. This made it easy to compute estimation matrices and controllers for each of the vertex models in the LPV system.

The scheduling parameters were assumed to be within the ranges $V_a \in [V_a^{min}, V_a^{max}]$ and $\gamma \in [\gamma^{min}, \gamma^{max}]$. The vertex models were set to correspond to each of the possible combinations of extremal values for the scheduling parameters. Corresponding estimators and controllers were computed for each of the vertex models.

It was proposed to use the measured airspeed V_a and flight angle γ by to compute the weighting coefficients $h(\boldsymbol{v}(t))$ by

$$
\begin{aligned}
h_1 &= \left(\frac{V_a^{max} - Va}{V_a^{max} - V_a^{min}}\right)\left(\frac{\gamma^{max} - \gamma}{\gamma^{max} - \gamma^{min}}\right), \quad \text{for model from } (V_a^{min}, \gamma^{min}) \\
h_2 &= \left(\frac{V_a^{max} - Va}{V_a^{max} - V_a^{min}}\right)\left(\frac{\gamma - \gamma^{min}}{\gamma^{max} - \gamma^{min}}\right), \quad \text{for model from } (V_a^{min}, \gamma^{max}) \\
h_3 &= \left(\frac{V_a - Va^{min}}{V_a^{max} - V_a^{min}}\right)\left(\frac{\gamma^{max} - \gamma}{\gamma^{max} - \gamma^{min}}\right), \quad \text{for model from } (V_a^{max}, \gamma^{min}) \\
h_4 &= \left(\frac{V_a - Va^{min}}{V_a^{max} - V_a^{min}}\right)\left(\frac{\gamma - \gamma^{min}}{\gamma^{max} - \gamma^{min}}\right), \quad \text{for model from } (V_a^{max}, \gamma^{max}).
\end{aligned}
\tag{5.1}
$$

This methodology applied directly with the LPV system did not work as expected. It was also attempted to alter the estimation algorithm to

$$
\begin{aligned}
\hat{\boldsymbol{x}}_t^j &= (I - H^j C^j)\hat{\boldsymbol{x}}_t^{-j} + K^j \boldsymbol{y}_t \\
\hat{\boldsymbol{x}}_{t+1}^{-j} &= A^j \hat{\boldsymbol{x}}_t^j + B^j \boldsymbol{u}_t
\end{aligned}
\tag{5.2}
$$

by removing the weighting within computation of the estimates for each vertex models. The weighting was instead added when computing the estimate from the LPV system:

$$\hat{\boldsymbol{x}}_t = \sum_{j=1}^{L} h(\boldsymbol{v}(t))\hat{\boldsymbol{x}}_t^j. \tag{5.3}$$

The proposed control algorithm remained the same. This did however not yield any positive results either.

Chapter 6

Discussion

6.1 Modelling

In the thesis a foundation has been made towards modelling and simulating a hybrid system convertiplane. Section (3.2) states that in order to verify the control strategies in simulation it is required to have accurate descriptions of the forces and moments working on the aircraft. In the following sections the most important forces and moments on the aircraft were described. The resulting model is essentially a fixed-wing aircraft with the added capability of generating lift force and rolling, pitching and yawing moments from the quadcopter propellers. This in many ways what would be expected to arrive at. However, there are many potential contributions that ought to be described that may affect the performance of the aircraft and controller that impacts the validity of the simulations. Many of these effects are well described in the literature both for fixed-wing aircraft and quadcopter aircraft. For conservative flight envelopes these effects can be neglected without much loss of performance.

During the transition phase from hovering to fixed-wing and back flight conditions are well beyond the normal scopes of the regular models. The airspeed is below stall speed for the aircraft and simply not high enough in order to generate the required lift to keep the aircraft in stable flight. At that point the quadcopter propellers will have to assist with generating lift and controlling the aircraft. Even if the airspeed is below stall velocity for the fixed-wing, it is still well above the normal working conditions of a quadcopter of this size. It was briefly mentioned in the paragraph regarding propeller aerodynamics that it is be expected to see a degradation in the effective thrust generated from the upward-facing quadcopter propellers, as well as blade-flapping effects becoming a considerable factor for large translational velocities. The analysis that was done with the *STARMAC* [20] is quite specific for that aircraft. Analysing similar effects on the Kestrel is probably a thesis on its own and well beyond the scope of this thesis. Another contribution that might affect

the aircraft during a transition mode might be the disruption of the airflow across the aerofoils from the quadcopter propellers. This will cause a loss of lifting force generated by the wings. None of these effects were investigated any further in this thesis, but might be of interest to analyse further in future work in order to create realistic simulation environments and develop more robust controllers.

6.2 Simulations

6.2.1 Noise

It was chosen to perform all simulations in this thesis in a deterministic setting, and not consider any process or measurement noise. The reason for this choice was to focus on the proposed solution method of using MMA and LPV methods for handling uncertainties in plant parameters and handling variations in flight conditions for the given systems. The advantage of using a deterministic setting is that it is easier to directly investigate the inherent robustness of the method, and made it possible to directly observe the effects of using linearised state-space systems for estimation and linear controllers. To include noise is be perfectly valid and the convergence proof given in [17] will still hold.

6.2.2 Quadcopter Simulations

In the simulations done for hovering with the quadcopter the full six degrees of freedom dynamic models were used as the system plant. Gravity and the forces and moments generated from the control inputs were the external forces considered.The control inputs were assumed to be direct force inputs affecting the UAV without going through computing the actual rotational velocities of each individual propeller, take into account their physical properties and then compute the effective forces and moments the propellers generate. This assumes that each pair of opposite facing propellers when generating roll, pitch and yaw moments will have to generate perfectly opposing thrusts for roll and pitch, while there has to be differences in the total angular velocities in order to generate yawing moments. This is not a realistic scenario.

In reality angular velocities will be unbalanced and there will also be a resulting yawing moment and linear forces arising from these asymmetries. It was assumed that these contributions were small and without a realistic physical model to work out from, creating physical models for the respective propellers would be an educated guess at best.

6.2.2.1 Hovering

Looking at Figure (5.3 c)) the quadcopter responds quickly and is able to track the sepoints accurately. A larger overshoot can be observed when the quadcopter attempts to track the second step change in the setpoint value at t =30, this is due to the integral term in the controller being smaller and not stabilised for the first setpoint change than the second thus creating a smaller overshoot. Without the integral term the controller was not able to handle the constant gravitational force and it resulted in a constant offset in the altitude p_d and vertical velocity w.

6.2.2.2 Setpoint Tracking

As seen in Figure (5.6) a) and b) the quadcopter is able to accurately track step responses in p_n and p_e directions with the overshoots seeming to be proportional to the magnitude of the step change. Figure (5.6) c) shows that the controller is able to stabilise the quadcopter at the desired setpoints. Some oscillations can be observed at points that corresponds to moments when there were changes in setpoints in the p_n and p_e directions. This is due to the quadcopter tilting to move to the new setpoint, which cause a loss in vertical lift thus a small loss in altitude and causing the quadcopter to have to stabilise in altitude once more. Figure(5.8) shows the Euler-angle positions and it can be seen that the controller is able to stabilise the quadcopter.

Figure (5.7) shows the dynamic weighting of each of the candidate models. The MMAE algorithm correctly identifies the candidate model with the parameter value most similar to the true parameter value in the plant. However, there are points at which the algorithm goes through a series of quick alternations between the different candidate models before stabilising again. Thes timing of these alternations correspond to the points when there are changes in setpoints in the p_n and p_e directions. These changes seem to occur instantly, but in fact the algorithm uses 20-30 iterations (0.2-0.3 seconds) going through continuous transitions. When the step changes in setpoint happen the quadcopter is forced away from its steady hovering state. It has to rotate and move, and those velocities can become quite substantial. When this occurs the nonlinear coupling parts of Equation(3.39) become prevalent, since these terms are not modelled into the linear observers it becomes difficult for the observers to correctly estimate the system state. Thus, leading to the algorithm making wrong decisions regarding which candidate model is the correct one since the specific value of the unknown parameter is less important for the system dynamics.

6.2.2.3 Trajectory Tracking

When set to follow continuous trajectories in p_n and p_e directions the controller manages to track the step transitions in altitude accurately in the same manner as the previous simulations. In Figures (5.9) a) and b) it can be seen that the controller is able to make the quadcopter track the desired trajectory correctly, but delayed by 1-2 seconds compared to the desired trajectory. One way of minimising this delay could have been to increase the weighting of the inertial positions in the LQR state weighting matrix Q_{LQR}, this did not yield substantially improved results. Too large weighting parameters resulted in general degradation of the controller performance.

In Figure (5.10) the controller is able to stabilise the Euler-angles satisfactorily. When following a slowly varying trajectory the quadcopter does not perform sharp manoeuvres which means the assumptions of small linear and angular velocities still hold for the linear observers. This can be confirmed by observing the dynamic weights in Figure (5.11) where the MMAE algorithm is able to correctly identify the candidate model with the m_{point} parameter value most resembling the true value in the plant, while the quadcopter is following a continuous trajectory.

6.2.2.4 Trajectory Tracking With Varying Weight

Figures (5.12 - 5.14) shows how the MMAE algorithm is able to precisely and correctly change between the candidate models during the simulation while the true value of m_{point} in the plant changes, even while the quadcopter is following a continuous trajectory. Thus confirming that it is indeed the unknown parameter that decides the candidate model of the MMA algorithm.

6.2.3 Aircraft Simulation

When simulating the longitudinal model of the aircraft it was chosen not to consider wind. This degrades the results validity compared to a realistic scenario. The choice was made in order to be able to focus on the performance of the controller and the MMA algorithm.

6.2.3.1 Single Model Simulation

By investigating Figures (5.16 - 5.18) it can be confirmed that the estimation is performing well and the controller is capable of maintaining level flight.

When the aircraft was set to follow desired trajectories in altitude and airspeed Figures (5.19 - 5.19) show that it tracked the trajectories accurately. The altitude plot shows some overshoot when switching between climb/descent and level

flight, and minor deviations during changes in the airspeed. The aircraft follows the desired airspeed trajectory accurately. When the aircraft changes flight path (altitude changes) small oscillations in the airspeed occur which it manages to stabilise from.

6.2.3.2 Multiple Model Simulation

Figures (5.22 - 5.22) show that the aircraft is controlled correctly while employing multiple models, and portray similar behaviour as when only considering a single model. As seen in Figure (5.25) the MMA algorithm correctly identifies the candidate model with $C_{L_\alpha} = 3.55$ which is the model with parameter value closest to the true parameter value $C_{L_\alpha} = 3.45$ in the plant.

Figure (5.26) shows the altitude and Figure (5.27) the airspeed when the aircraft was set to follow a trajectory. It behaves similarly to the single model simulation. However, looking at Figure (5.29) the dynamic weights are a mess. For time 0 until $\sim$30 the aircraft is climbing and the MMA algorithm chooses the model with $C_{L_\alpha} = 2.55$. From time $\sim$50 to $\sim$75, the plane is changing airspeed, and during that period it oscillate between different candidate models. From time 75 until 175 the aircraft flies at constant altitude while maintaining an airspeed $V_a = 30m/s$ and the MMA algorithm chooses the model with $C_{L_\alpha} = 4.55$. From time 175 until 210 the aircraft changes altitude and desired airspeed, and the MMA algorithm jumps between different candidate models and briefly settles on the candidate model with $C_{L_\alpha} = 4.55$. First after time 250 when airspeed has stabilised on the desired airspeed and the plane flies at a stable altitude the MMA algorithm correctly identifies the correct model with $C_{L_\alpha} = 3.55$ which is the one closest to the true value of the plant which is $C_{L_\alpha} = 3.45$.

This seeming inability for the MMA algorithm to decide on the correct candidate model is most likely a result of the process of designing estimators and controllers. Each of the estimators and corresponding controllers were designed based on the calculated trim conditions for the various candidate models. Trim conditions for each candidate model were computed to follow the same airspeed and constant altitude. Comparing the computed the trim state for the candidate model with the lowest and highest C_{L_α}-values; the trim state for the candidate model corresponding to the lowest $C_{L_\alpha} = 2.55$ is $\boldsymbol{x}^* = [0, 24.85, 2.69, 0.1077, 0]$, and the candidate model with the largest $C_{L_\alpha} = 4.55$ is $\boldsymbol{x}^* = [0, 24.95, 1.59, 0.06, 0]$, the other candidate models had trim states with values that were distributed between these two extremes.

During ascent the aircraft will require to increase its pitch angles, and effectively angle of attack. For descent it will have to decrease its pitch angle. This can be confirmed in Figure(5.30). Based on this it is reasonable to assume that the estimator which is computed for the candidate model with the largest trim pitch angle will perform better during ascent, and that the controller which controls the attitude towards the larges pitch will perform better during ascent. This

is exactly what happens from time 0 to 30 where the aircraft is climbing and the MMA algorithm chooses the candidate model with the largest trim pitch angle. The same argument can be used for time 200 to 220 where the aircraft is descending and the MMA algorithm chooses the model with the lowest trim pitch angle. For time 75 to 175 where the aircraft is set to maintain a higher airspeed and constant altitude. In order to maintain a higher airspeed and at constant altitude the aircraft will have to decrease pitch angle, which can be seen in Figure (5.30). Thus the MMA algorithm chooses the model with the lowest trim pitch angle.

For the multiple model adaptive estimation and control simulations of the fixed-wing aircraft it was chosen to use C_{L_α} as the unknown parameter instead of adding an unknown weight in a similar manner done for the quadcopter simulations. The same effects of the trim conditions effectively being the deciding factor between the candidate models could be seen when applying an unknown weight. However, this effect was more pronounced when varying C_{L_α} than varying the weight since variations in C_{L_α} had larger impact on the computed trim state pitch angle θ^*.

6.2.3.3 LPV Implementation

As explained previously it is expected that LPV estimation and control could have been a good method for fixing the issues that arose when applying the MMA algorithm to the fixed-wing aircraft. An attempt was made at implementing an LPV architecture onto the MMA method for the fixed-wing aircraft. This did not yield any positive results. Section (5.3.3) describes the approach taken towards constructing the LPV system. The approach taken was heavily influenced by [38].

The system can be described at any point of the state $\boldsymbol{x} = [-, u, w, \theta, q]$ using the non-linear embedding of the parameters approach described in Section (4.3.3.3). It was attempted without success to construct an LQR controller based on these state-space systems. Utilising the method of system augmentation in Section (4.3.4) by adding integral states directly on this state-space system yielded an uncontrollable system.

As a workaround it was chosen to instead use the linearised state-space model from Section (4.3.3.2) for which the integral action LQR controller was initially designed [45][1]. However, this requires the computation of trim conditions on which to base the linearisation method. There are infinitely many ways of computing trim conditions based on what is desired to achieve. In this thesis a method was created which computed the trim conditions given a desired airspeed and flight path angle. This seemed like a valid approach based on the desire to follow trajectories in altitude and airspeed. The calculated trim conditions were then used to compute the state-space matrices used for

[1] The web-page by the authors of [35] which contains an unpublished supplement to the book.

LPV estimation, thus local observers and corresponding linear controllers were obtained to work as vertices in the LPV system. It should be noted that each of the these vertex systems were capable of controlling the aircraft during stable flight around its trim state.

Since the trim conditions for each of the vertex systems were computed for a given airspeed and flight path angle they were deemed a natural choice for computing the weighting between the vertex systems based on the estimated states of the aircraft. This could have been a valid choice, however not with the proposed method for computing the weighting coefficients in Equation (5.1). This approach assumes a linear relationship between the vertex systems. Using the same approach for computing the weights with only a single scheduling parameter the two weights h_1 and h_2 would be related by $h_1 = 1 - h_2$. Looking at how the vertex systems are added together to create the polytopic LPV system in Equation (3.72), the same dependence on the scheduling parameter and relationship between the weighting coefficients would be expected to see between the matrices of the vertex systems. Looking at Table (4.3) such a linear relationship is clearly not applicable.

The problem outline above regarding the proposed LPV architecture might not be the complete explanation for the failure to implement an LPV system. It does however reveal clear issues with the attempted approach, and contradictions with the required assumptions for regarding the LPV system as polytopic over the scheduling parameters. Given the restricted time nature of the thesis this flaw was discovered too late to come up with and implement an alternative approach.

6.3 Control Methodology

Utilising a linear controller to control a non-linear plant is inherently limited by its linearity. Wander too far off from the linearisation point and the robustness of the controller quickly deteriorates. Throughout the thesis the LQR controller was used for both the quadcopter and the fixed-wing aircraft, and for the given settings in the thesis it performed well. Its ease of implementation made it possible to quickly employ it to a wide range of different systems while simultaneously guaranteeing optimality for the given state-space system and weighting matrices. With the added integral states the controller was able to track the desired setpoints and trajectories in a sufficient manner.

Given that the LQR controller worked this well for both controlling the quadcopter and fixed-wing systems it should be expected that the same approach should be applicable to a transition phase between the two stages. A proposed method would be to extend the control algorithm of the longitudinal fixed-wing flight to include vertical thrust and pitching moment coming from the quadcopter propellers in order to assist lifting and control until stable fixed-wing flight is achievable.

In this thesis the controllers of the MMA systems were all designed in the same manner. With the control input being computed as the linear combination of all the control inputs from the various candidate models weighted by their dynamic weights, it was always assumed that essentially any of the controllers of the candidate models should be able to control the system and that choosing the wrong candidate model would not cause fatal outcomes. As described in [6], multiple model adaptive control can be utilised for vastly different controllers under varying flight conditions. However, then a mechanism must be in place in order to ensure that when switching to a new candidate model the controller will be able to control the system in a sufficient manner. If a complete estimation and control suite using the MMA algorithm were to be used for the whole flight envelope of the hybrid UAV system, secure switching mechanisms would have to be employed during the transition phases.

6.4 MMA and LPV Methods

A huge advantage of using the MMA and LPV paradigms is the ability of estimating and controlling a highly nonlinear system given limited knowledge of the model parameters and possibility of highly varying flight conditions.

The MMA algorithm has shown its ability to correctly identify the optimal candidate model given a system with large uncertainties in predetermined plant parameters. Utilising linear observers and linear controllers the algorithm worked well as long as the unmodelled dynamics did not become too dominant. The problems with identifying the correct model when the fixed-wing aircraft was set to track a trajectory were more likely due to the methodology for estimator and controller design than the MMA method. Parts of that problem could have been alleviated with a successful implementation of the LPV methodology.

Even though the attempt at implementing LPV estimation and control was unsuccessful it is still considered a strong candidate towards ensuring stable flight through the complete flight envelope of Kestrel. The features of the method are considered well suited towards handling the challenges of transition and fixed-wing flight, and the methodology fits well together with the MMA method.

The multiple model algorithm is a simple yet effective way of working with a dynamical system operating under large parameter uncertainties. In theory the concept could be extended to concern a large set of parameters in the plant while having the algorithm sort things out by finding the optimal parameter set. However, as the number of unknown parameters increase, the computational complexity of the algorithm grows exponentially. With the approach taken in this thesis by using N fixed candidate models for each unknown parameter, with M unknown parameters the algorithm is required to compute estimates and weighting of N^M models at each iteration. Adding the LPV system with L

vertex systems for each model the number of models to compute increases to $L \cdot N^M$. With the often limited computing power found on board a UAV, and the hard real-time requirement of an autopilot working at high frequencies it is safe to say that the capabilities are limited. One solution could be to optimise the unknown parameters in succession in order to relieve the computational requirement at each iteration. On top of everything comes the ever present problem in optimisation of local extrema.

6.5 Further Work

It is expected that the development of the Kestrel concept will continue at FFI.

In this thesis the dual system convertiplane hybrid UAV concept has for the first time been investigated as a complete system in an organised manner. A solution concept of modelling, estimation, parameter identification and control has been presented. Due to the inherent complexity of the problem and the limitations of the thesis every segment of the solution concept are open for improvement.

The most direct points of continuation on the work in this thesis would be to improve on the shortcomings of the achieved results. A successful implementation of the LPV system would make it possible to further analyse the validity of using the MMA algorithm together with the proposed methods for estimation and control of the fixed-wing aircraft. Extending the simulations of the fixed-wing aircraft to include the full 6 DoF dynamics of the aircraft and design a controller for the lateral dynamics is required in order to achieve flight of the complete aircraft. Introducing noise and wind to the simulations would further work towards confirming the validity of the methods.

The biggest issue that was not addressed directly in this thesis is the transition phases between quadcopter and fixed-wing mode. Investigating the dynamics of the aircraft during this phase, and specifically for the Kestrel, would make it possible to build a more realistic simulation environment and provide valuable information regarding the possible challenges when designing more robust controllers. With the presented solution concept of MMA LPV estimation and control having good track records for controlling both fixed-wing and quadcopter aircraft, it would be interesting to see a successful implementation of the LPV methodology and extend these concepts into the transition regime by combining the capabilities of the quadcopter with the fixed-wing flight. Of special interest would be to see how the quadcopter control interacts with the fixed-wing dynamics.

Implementing the methods outlined in this thesis in the physical Kestrel are not straightforward either. As described earlier, it is not feasible to implement the MMA algorithm for all the parameters of the aircraft and let the algorithm sort things out on its own. The MMA algorithm is best employed to a few selected parameters that happen to change on the spot or throughout an experiment

that are not easily determined without a more careful examination, or other factors that are not easily determined. A series of measurements must be performed in order to determine the physical parameters of the aircraft which are needed for estimators and controllers as well as for building a robust simulation environment.

Chapter 7

Conclusion

In this thesis modelling, estimation and control of a dual system convertiplane has been investigated.

A mathematical model for a dual system convertiplane has been developed based on the concept plane Kestrel. The model consist of twelve equations constituting the full six degrees of freedom dynamic model for the aircraft. It includes propulsion forces and actuator deflections from both the fixed-wing aircraft and the quadcopter propellers. The external forces and moments that are expected to have the largest contributions on the aircraft dynamics have been investigated and been included in the model.

The system is highly non-linear and the controller will have to be able to handle a plethora of different flight conditions. Everything from hovering using the quadcopter propellers, maintaining steady flight through a transition phase to fixed-wing flight, maintaining stable fixed-wing flight and the transition back to quadcopter mode. All of this while also considering possibly large uncertainties in the parameters of the aircraft model. A linear parameter varying multiple model adaptive architecture was suggested as a possible solution method.

Two simulation models were constructed. One of a quadcopter during hover, and the other of the longitudinal flight of a fixed-wing aircraft. The models were constructed with predefined unknown parameters. Linearised systems for estimation and control were obtained for both systems. The MMA algorithm together with and infinite horizon LQR controller with integral action was employed in order to estimate and control the systems while simultaneously identify the unknown parameters.

The quadcopter was successfully controlled in a series of simulations following various trajectories and changes in desired setpoints while an unknown mass was attached to it which changes its dynamics. The multiple model algorithm was able to dynamically correctly identify the candidate model which had the value of the weight which most closely resembled the true weight on the quadcopter. This worked well as long as state of the quadcopter remained

close to the linearisation point for the linearised state-space system and corresponding controller. With too large deviations from the linearisation point the multiple model algorithm struggled with identifying the correct model, but was able to correct itself once stability was regained.

The Longitudinal flight of a fixed-wing aircraft was simulated. Successful flight while following a trajectory in altitude and desired airspeed was achieved. This was done with an state estimator using embedding of the non-linearities at a calculated trim point, and LQR controller with integral action obtained by using the linearised state-space matrices calculated at the same trim point. With uncertainties in the C_{L_α} aerodynamic parameter the multiple model algorithm was able to correctly identify the candidate model with the suggested parameter value which most closely resembled the true parameter value of the aircraft while maintaining stable flight at the trim point. While following a trajectory in altitude and desired airspeed the multiple model algorithm was not able to identify the correct model during flight conditions that deviated too far from the conditions the trim point was calculated for. During these regimes other factors of estimator and controller design seemed to determine the quality of the models.

Implementation of a linear parameter varying method was attempted without success. Despite this, the linear parameter varying methodology is still considered a strong candidate towards handling the huge variations in flight conditions the Kestrel is expected to meet. Implementation of the linear parameter varying architecture and a more robust controllers are considered the best approaches for further work.

The work done in this thesis is an important step in enabling further development of the Kestrel as a platform and concept.

Bibliography

[1] Panos J. Antsaklis and Anthony N. Michel. *A Linear Systems Primer*. Springer Science & Business Media, 3rd Dec. 2007. 524 pp. ISBN: 978-0-8176-4661-5.

[2] *ArduPilot Open Source Autopilot*. URL: http://ardupilot.org/ (visited on 04/05/2019).

[3] M. E. Argyle, R. W. Beard and S. Morris. 'The Vertical Bat Tail-Sitter: Dynamic Model and Control Architecture'. In: *2013 American Control Conference*. 2013 American Control Conference. June 2013, pp. 806–811. DOI: 10.1109/ACC.2013.6579935.

[4] Karl J. Åström and Tore Hägglund. 'Advanced Pid Control'. In: *The Instrumentation, Systems, and Automation Society* (2006).

[5] Karl J. Åström and Björn Wittenmark. *Adaptive Control: Second Edition*. Courier Corporation, 26th Apr. 2013. 596 pp. ISBN: 978-0-486-31914-8.

[6] Michael Athans, Sajjad Fekri and Antonio Pascoal. 'ISSUES ON ROBUST ADAPTIVE FEEDBACK CONTROL'. In: *IFAC Proceedings Volumes*. 16th IFAC World Congress 38.1 (1st Jan. 2005), pp. 547–577. ISSN: 1474-6670. DOI: 10.3182/20050703-6-CZ-1902.01036. URL: http://www.sciencedirect.com/science/article/pii/S1474667016370483 (visited on 04/05/2019).

[7] M. Athans et al. 'The Stochastic Control of the F-8C Aircraft Using a Multiple Model Adaptive Control (MMAC) Method–Part I: Equilibrium Flight'. In: *IEEE Transactions on Automatic Control* 22.5 (Oct. 1977), pp. 768–780. ISSN: 0018-9286. DOI: 10.1109/TAC.1977.1101599.

[8] Randal W Beard. 'Quadrotor Dynamics and Control'. In: (19th Feb. 2008), p. 48.

[9] *Bell Boeing V-22 Osprey*. In: *Wikipedia*. Page Version ID: 891904139. 10th Apr. 2019. URL: https://en.wikipedia.org/w/index.php?title=Bell_Boeing_V-22_Osprey&oldid=891904139 (visited on 04/05/2019).

[10] *Bell Eagle Eye*. In: *Wikipedia*. Page Version ID: 875812640. 29th Dec. 2018. URL: https://en.wikipedia.org/w/index.php?title=Bell_Eagle_Eye&oldid=875812640 (visited on 04/05/2019).

[11] Yolanda Bolea and Vicenç Puig. 'Gain-Scheduling Multivariable LPV Control of an Irrigation Canal System'. In: *ISA Transactions* 63 (1st July 2016), pp. 274–280. ISSN: 0019-0578. DOI: 10.1016/j.isatra.2016.03.009. URL: http://www.sciencedirect.com/science/article/pii/S0019057816300325 (visited on 25/05/2019).

[12] Corentin Briat. *Linear Parameter-Varying and Time-Delay Systems*. Vol. 3. Advances in Delays and Dynamics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. ISBN: 978-3-662-44049-0 978-3-662-44050-6. DOI: 10.1007/978-3-662-44050-6. URL: http://link.springer.com/10.1007/978-3-662-44050-6 (visited on 17/05/2019).

[13] Ferit Çakici and M. Kemal Leblebicioğlu. 'Control System Design of a Vertical Take-off and Landing Fixed-Wing UAV'. In: *IFAC-PapersOnLine*. 14th IFAC Symposium on Control in Transportation SystemsCTS 2016 49.3 (1st Jan. 2016), pp. 267–272. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2016.07.045. URL: http://www.sciencedirect.com/science/article/pii/S2405896316302415 (visited on 08/01/2019).

[14] E. Cetinsoy et al. 'Design and Construction of a Novel Quad Tilt-Wing UAV'. In: *Mechatronics*. Special Issue on Intelligent Mechatronics (LSMS2010 & ICSEE2010) 22.6 (1st Sept. 2012), pp. 723–745. ISSN: 0957-4158. DOI: 10.1016/j.mechatronics.2012.03.003. URL: http://www.sciencedirect.com/science/article/pii/S095741581200044X (visited on 04/05/2019).

[15] Richard C. Dorf and Robert H. Bishop. *Modern Control Systems*. 12 edition. Prentice Hall: Pearson, 29th July 2010. 1104 pp. ISBN: 978-0-13-602458-3.

[16] William Ellis et al. 'Design, Build and Test of the VTOL UAV-'Kestrel''. Mastes Thesis. University of Southampton.

[17] V. Hassani et al. 'Multiple Model Adaptive Estimation and Model Identification Usign a Minimum Energy Criterion'. In: *2009 American Control Conference*. 2009 American Control Conference. June 2009, pp. 518–523. DOI: 10.1109/ACC.2009.5160446.

[18] *History of Quadcopters and Multirotors*. URL: https://www.krossblade.com/history-of-quadcopters-and-multirotors (visited on 04/05/2019).

[19] Menno Hochstenbach et al. 'Design and Control of an Unmanned Aerial Vehicle for Autonomous Parcel Delivery with Transition from Vertical Take-off to Forward Flight – VertiKUL, a Quadcopter Tailsitter'. In: *International Journal of Micro Air Vehicles* 7.4 (1st Dec. 2015), pp. 395–405. ISSN: 1756-8293. DOI: 10.1260/1756-8293.7.4.395. URL: https://doi.org/10.1260/1756-8293.7.4.395 (visited on 04/05/2019).

[20] Gabriel Hoffmann et al. 'Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment'. In: *AIAA Guidance, Navigation and Control Conference and Exhibit*. AIAA Guidance, Navigation and Control Conference and Exhibit. Hilton Head, South Carolina: American Institute of Aeronautics and Astronautics, 20th Aug. 2007. ISBN: 978-1-62410-015-4. DOI: 10.2514/6.2007-6461. URL: http://arc.aiaa.org/doi/10.2514/6.2007-6461 (visited on 23/01/2019).

[21] M. Hua et al. 'Introduction to Feedback Control of Underactuated VTOLvehicles: A Review of Basic Control Design Ideas and Principles'. In: *IEEE Control Systems Magazine* 33.1 (Feb. 2013), pp. 61–75. ISSN: 1066-033X. DOI: 10.1109/MCS.2012.2225931.

[22] *Hybrid Quadrotor - Hybrid Quadcopter - VTOL UAV - Autonomous Takeoff - Autonomous Landing*. URL: https://www.latitudeengineering.com/products/hq/ (visited on 24/05/2019).

[23] Petros A. Ioannou and Jing Sun. *Robust Adaptive Control*. Courier Corporation, 19th Dec. 2012. 850 pp. ISBN: 978-0-486-49817-1.

[24] *JUMP 20 | Arcturus UAV*. URL: https://arcturus-uav.com/product/jump-20 (visited on 04/05/2019).

[25] king-theme.com. *Transition*. 1st Nov. 2017. URL: https://www.altiuas.com/transition/ (visited on 04/05/2019).

[26] Francisco Ronay López-Estrada et al. 'LPV Model-Based Tracking Control and Robust Sensor Fault Diagnosis for a Quadrotor UAV'. In: *Journal of Intelligent & Robotic Systems* 84.1 (1st Dec. 2016), pp. 163–177. ISSN: 1573-0409. DOI: 10.1007/s10846-015-0295-y. URL: https://doi.org/10.1007/s10846-015-0295-y (visited on 25/05/2019).

[27] R. Mahony, V. Kumar and P. Corke. 'Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor'. In: *IEEE Robotics Automation Magazine* 19.3 (Sept. 2012), pp. 20–32. ISSN: 1070-9932. DOI: 10.1109/MRA.2012.2206474.

[28] Hanmant G. Malkapure and M. Chidambaram. 'Comparison of Two Methods of Incorporating an Integral Action in Linear Quadratic Regulator'. In: *IFAC Proceedings Volumes*. 3rd International Conference on Advances in Control and Optimization of Dynamical Systems (2014) 47.1 (1st Jan. 2014), pp. 55–61. ISSN: 1474-6670. DOI: 10.3182/20140313-3-IN-3024.00105. URL: http://www.sciencedirect.com/science/article/pii/S1474667016326350 (visited on 26/05/2019).

[29] Robert G McSwain, Louis J Glaab and Colin R Theodore. 'Greased Lightning (GL-10) Performance Flight Research – Flight Data Report'. In: (), p. 79.

[30] K. S. Narendra and J. Balakrishnan. 'Adaptive Control Using Multiple Models'. In: *IEEE Transactions on Automatic Control* 42.2 (Feb. 1997), pp. 171–187. ISSN: 0018-9286. DOI: 10.1109/9.554398.

[31] Ugur Ozdemir et al. 'Design of a Commercial Hybrid VTOL UAV System'. In: *Journal of Intelligent & Robotic Systems* 74.1-2 (Apr. 2014), pp. 371–393. ISSN: 0921-0296, 1573-0409. DOI: 10.1007/s10846-013-9900-0. URL: http://link.springer.com/10.1007/s10846-013-9900-0 (visited on 04/05/2019).

[32] PixhawkAdmin. *Home Page*. URL: http://pixhawk.org/ (visited on 04/05/2019).

[33] Vasile Prisacariu. 'THE HISTORY AND THE EVOLUTION OF UAVs FROM THE BEGINNING TILL THE 70s'. In: *Journal of Defense Resources Management (JoDRM)* 8.1 (2017), pp. 181–189. ISSN: 2068-9403. URL: https://www.ceeol.com/search/article-detail?id=544698 (visited on 04/05/2019).

[34] Quan Quan. *Introduction to Multicopter Design and Control*. Singapore: Springer Singapore, 2017. ISBN: 978-981-10-3381-0 978-981-10-3382-7. DOI: 10.1007/978-981-10-3382-7. URL: http://link.springer.com/10.1007/978-981-10-3382-7 (visited on 06/05/2019).

[35] Randal W. Beard and Timothy W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton, N.J: Princeton University Press, 2012. xiii+300. ISBN: 978-0-691-14921-9.

[36] D. Rotondo, V. Hassani and A. Cristofaro. 'A Multiple Model Adaptive Architecture for the State Estimation in Discrete-Time Uncertain LPV Systems'. In: *2017 American Control Conference (ACC)*. 2017 American Control Conference (ACC). May 2017, pp. 2393–2398. DOI: 10.23919/ACC.2017.7963311.

[37] Damiano Rotondo et al. 'Automated Generation and Comparison of Takagi–Sugeno and Polytopic Quasi-LPV Models'. In: *Fuzzy Sets and Systems*. Theme: Fuzzy Systems 277 (15th Oct. 2015), pp. 44–64. ISSN: 0165-0114. DOI: 10.1016/j.fss.2015.02.002. URL: http://www.sciencedirect.com/science/article/pii/S0165011415000652 (visited on 25/05/2019).

[38] Damiano Rotondo et al. 'Icing Diagnosis in Unmanned Aerial Vehicles Using an LPV Multiple Model Estimator'. In: *IFAC-PapersOnLine*. 20th IFAC World Congress 50.1 (1st July 2017), pp. 5238–5243. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2017.08.462. URL: http://www.sciencedirect.com/science/article/pii/S2405896317308248 (visited on 27/05/2019).

[39] Damiano Rotondo et al. 'LPV Model Reference Control for Fixed-Wing UAVs'. In: *IFAC-PapersOnLine*. 20th IFAC World Congress 50.1 (1st July 2017), pp. 11559–11564. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2017.08.1640. URL: http://www.sciencedirect.com/science/article/pii/S2405896317322413 (visited on 08/01/2019).

[40] Francesco Sabatino. *Quadrotor Control: Modeling, Nonlinearcontrol Design, and Simulation*. 2015. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-175380 (visited on 25/05/2019).

[41] Adnan S. Saeed et al. 'A Survey of Hybrid Unmanned Aerial Vehicles'. In: *Progress in Aerospace Sciences* 98 (1st Apr. 2018), pp. 91–105. ISSN: 0376-0421. DOI: 10.1016/j.paerosci.2018.03.007. URL: http://www.sciencedirect.com/science/article/pii/S0376042117302233 (visited on 04/05/2019).

[42] Jeff S. Shamma. 'An Overview of LPV Systems'. In: *Control of Linear Parameter Varying Systems with Applications*. Ed. by Javad Mohammadpour and Carsten W. Scherer. Boston, MA: Springer US, 2012, pp. 3–26. ISBN: 978-1-4614-1833-7. DOI: 10.1007/978-1-4614-1833-7_1. URL: https://doi.org/10.1007/978-1-4614-1833-7_1 (visited on 25/05/2019).

[43] Dan Simon. 'Kalman Filtering for Fuzzy Discrete Time Dynamic Systems'. In: *Applied Soft Computing* 3.3 (1st Nov. 2003), pp. 191–207. ISSN: 1568-4946. DOI: 10.1016/S1568-4946(03)00034-6. URL: http://www.sciencedirect.com/science/article/pii/S1568494603000346 (visited on 25/05/2019).

[44] C. Sloth, T. Esbensen and J. Stoustrup. 'Active and Passive Fault-Tolerant LPV Control of Wind Turbines'. In: *Proceedings of the 2010 American Control Conference*. Proceedings of the 2010 American Control Conference. June 2010, pp. 4640–4646. DOI: 10.1109/ACC.2010.5531061.

[45] *Start | Small Unmanned Aircraft: Theory and Practice*. URL: http://uavbook.byu.edu/doku.php (visited on 25/05/2019).

[46] Brian L. Stevens, Frank L. Lewis and Eric N. Johnson. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. John Wiley & Sons, 2nd Nov. 2015. 768 pp. ISBN: 978-1-118-87098-3.

[47] Seunghee Yu and Yongjin Kwon. 'Development of VTOL Drone for Stable Transit Flight'. In: *Journal of Computer and Communications* 05 (9th May 2017), p. 36. DOI: 10.4236/jcc.2017.57004. URL: https://www.scirp.org/journal/PaperInformation.aspx?PaperID=76198&#abstract (visited on 04/05/2019).