DESIGN AND CONTROL OF A TILT-ROTOR UNMANNED AERIAL VEHICLE FOR WETLAND INSPECTION

by

Yih Tang Yeo

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science
University of Toronto Institute for Aerospace Studies
University of Toronto

© Copyright 2018 by Yih Tang Yeo
Abstract

Design and Control of A Tilt-Rotor Unmanned Aerial Vehicle for Wetland Inspection

Yih Tang Yeo
Master of Applied Science
University of Toronto Institute for Aerospace Studies
University of Toronto
2018

This thesis presents a tilt-rotor unmanned aerial vehicle (UAV) that combines a quadrotor frame and a tailless wing-body through a servo-operated tilting mechanism. The UAV is capable of vertical take-off and landing (VTOL) on land and water, has high endurance and long range. The dynamics of the UAV are modelled using a built-up method with the motor-propeller data, the aerodynamic analysis data, and the computer-aided design (CAD) modelling data. For transition from hover to forward flight, a gain-scheduled altitude controller is developed to ensure minimal altitude change, and a singularity-free quaternion-based attitude tracking controller is implemented to command the pitch angle of the quadrotor-body from $0^\circ$ to $-90^\circ$. The dynamics and controllers are simulated using MATLAB Simulink to show successful transition. The altitude controller proposed is implemented in the ArduPilot firmware, and flight tests are conducted to validate the flight-worthiness of the proposed design and control.
Acknowledgements

I would like to thank and express my utmost appreciation and gratitude to Dr. Hugh H. T. Liu, who has supervised this research for the past two years. This thesis would not have been possible without his advice, guidance, and support. This gratitude is extended to my research assessment committee (RAC) which includes Professor Peter Grant and Professor Chris Damaren. I would also want to take this opportunity to thank my parents, siblings, and extended family members abroad for being ever so supportive to me through thick and thin.

I would also like to acknowledge the help I have received from my wonderful colleagues in the Flight Systems and Control (FSC) Group at the University of Toronto Institute for Aerospace Studies (UTIAS) – Dawei Li, Blago Hristovski, Silas Graham, Joshua Furtado, Longhao Qian, and Qingrui Zhang – for their technical inputs and suggestions throughout the formulation of the UAV design, as well as for their assistance during flight operations. This appreciation is also extended to my friends for the insightful discussions and support they provided.

The development and flight testing of the unmanned aerial vehicle (UAV) would not have been possible without the financial support from the Queen’s University Department of Biology, in addition to the funding from the Natural Sciences and Engineering Research Council (NSERC) Collaborative Research and Training Experience (CREATE) Program and the NSERC Discovery Grants Program. I would also like to thank Professor Peter Grant for providing the student design lab facility at UTIAS for fabrication of the UAV, as well as John Stinchcombe and Stephan Schneider for their help in coordinating use of the open fields in Kofflers Scientific Reserve (KSR) at Jokers Hill for flight testing of the AmphiQuad UAV.

Last but not least, I would like to thank the University of Toronto Aerospace Team (UTAT) for inspiring me and providing me the opportunity to gain practical knowledge and skills in aircraft design, as well as the understanding of the inner workings of an unmanned aerial system (UAS). Being a part of the team since 2012, I was given precious opportunities to lead the team and participate in design competitions, which exposed me to the technical details of every component on a UAV, novel construction techniques and design analysis tools, as well as the experience of planning and conducting flight operations.
# Contents

Acknowledgements iii

Table of Contents iv

Nomenclature vi

1 Introduction 1
   1.1 Background and Motivation ................................................. 1
   1.2 Related Work ........................................................................... 2
   1.3 Research Objectives and Goals ................................................. 7

2 Design of the AmphiQuad UAV 9
   2.1 Design Tools ............................................................................ 9
   2.2 Airfoil and Wing Design ............................................................ 11
   2.3 Quadrotor Frame Design ............................................................ 14
   2.4 Tilt-Rotor Interface Design ......................................................... 15
   2.5 Wing Sections Interface Design ................................................ 17
   2.6 Propulsion Selection and Analysis ............................................. 18
   2.7 Construction Techniques and Material Selection ......................... 19
   2.8 Avionics Selection ................................................................. 22
   2.9 Mass Breakdown ..................................................................... 23
   2.10 Performance Validation ............................................................ 24
   2.11 Summary of AmphiQuad UAV Design ....................................... 26

3 Dynamics of the AmphiQuad UAV 28
   3.1 Reference Frames ..................................................................... 28
   3.2 Equations of Motion ................................................................. 30
   3.3 Center of Gravity (C.G.) and Combined Moment of Inertia ............ 31
   3.4 Contribution of Forces and Moments ........................................ 33

4 Control of the AmphiQuad UAV 37
   4.1 Gain-Scheduled Altitude Controller ......................................... 37
4.2 Quaternion-based Attitude Controller ........................................... 50

5 Firmware Development ................................................................. 54
  5.1 ArduPilot Repository and Structure ........................................... 54
  5.2 Choice of Firmware and Autopilot Mounting Location ..................... 56
  5.3 Transition Between Hover and Forward Flight ................................. 57
  5.4 List of Modifications Made to the Firmware ................................... 59

6 Simulation ..................................................................................... 61
  6.1 MATLAB Simulink Model ........................................................... 61
  6.2 Simulation Results ..................................................................... 62

7 Experimental Work ........................................................................ 67
  7.1 Development and Flight Testing of Test Frame ................................. 67
  7.2 Flight Testing of AmphiQuad UAV ............................................... 73
  7.3 Suggestions for Improvements ..................................................... 78

8 Conclusions .................................................................................. 81

A Online Resources and Repositories ................................................. 83
  A.1 AVL Files .................................................................................. 83
  A.2 ArduPilot Firmware ................................................................. 83
  A.3 ArduPilot Documentation .......................................................... 84

B Aerodynamics Coefficients ............................................................. 85

Bibliography .................................................................................... 86
Nomenclature

Abbreviations
C.G. Center of Gravity
CAD Computer Aided Design
CFD Computational Fluid Dynamics
CF Carbon Fiber
COTS Commercial Off-The-Shelf
DI Dynamic Inversion
DOF Degree Of Freedom
ESC Electronic Speed Controller
GCS Ground Control Station
GPS Global Positioning System
KSR Koffler Scientific Reserve
LE Leading Edge
LQR Linear Quadratic Regulator
MIT Massachusetts Institute of Technology
NACA National Advisory Committee for Aeronautics
NASA National Aeronautics and Space Administration
NN Neural Network
PID Proportional–Integral–Derivative
PP Pole Placement
PWM Pulse Width Modulation
QUBS Queen’s University Biological Station
TECS Total Energy Control System
UAS Unmanned Aerial System
UAV Unmanned Aerial Vehicle
VTOL Vertical Take-Off and Landing
VT Vertical Tail

Subscripts
$B$ Wing-body reference frame
$D$ Desired attitude reference frame
\[ \mathcal{E} \] Earth (inertial) reference frame
\[ \mathcal{Q} \] Quadrotor-body reference frame
\[ \mathcal{W} \] Wind reference frame
0 Equilibrium value
e Error
T Vertical Tail (Quadrotor Frame)
W Wing-body
ax Axial direction
D Drag
d, req Desired value
FB Feedback
FF Feedforward
L Lift
m Pitching moment
x, y, z Wing-body coordinate axes
lim Limit

Symbols
\[ \alpha \] Angle of attack
\[ \bar{c} \] Mean aerodynamic chord
\[ \beta \] Sideslip angle
\[ \delta_e \] Elevon deflection angle
\[ \delta_T \] Throttle output
\[ \gamma \] Servo tilt angle (0° for VTOL mode, 90° for fixed-wing mode)
\[ \gamma_{\text{path}} \] Flight path angle
\[ \Omega \] Angular velocity of propellers
\[ \Phi \] Quaternion rotation angle
\[ \phi, \theta, \psi \] Roll, Pitch, Yaw angles
\[ \rho \] Air density
\[ \sigma \] Gain for equilibrium speed update
\[ \tilde{\mathcal{F}} \] Coordinate reference frame
\[ b \] Span of lifting surface
\[ C \] Aerodynamic coefficients
\[ c \] Chord length
\[ g \] Gravitational acceleration
\[ h \] Normalized length to the mean aerodynamic chord
\[ J \] Cost function (of LQR and planform optimization)
\[ k \] Controller gain
\[ k_m \] Motor constant (moment-to-thrust)
\[ l \] Length
$L, M, N$ Moments in the body axes  
$m$ Mass  
$p, q, r$ Body angular rates  
$S$ Surface area of lifting surface  
$T$ Motor thrust  
$t$ Thickness  
$V$ Airspeed / Speed  
$W$ Weight  
$X, Y, Z$ Forces in the body axes

**Matrices**

$C$ Rotation matrix $\in SO(3)$  
$I$ Inertia matrix $\in \mathbb{R}^{3 \times 3}$ calculated at C.G.  
$J$ Inertia matrix $\in \mathbb{R}^{3 \times 3}$  
$K$ Controller gain matrix $\in \mathbb{R}^{3 \times 3}$  
$Q$ Symmetric Positive Definite Matrix for LQR; weights of the states  
$R$ Symmetric Positive Definite Matrix for LQR; weights of the control outputs

**Superscripts**

$\times$ Cross product in matrix form  
$T$ Transpose of matrix or vector

**Vectors**

$\Gamma$ Inertia vector parameterization $\in \mathbb{R}^{6 \times 1}$  
$\omega$ Angular velocity $\in \mathbb{R}^{3 \times 1}$  
$\tau$ Torque $\in \mathbb{R}^{3 \times 1}$  
$\epsilon, \eta$ Quaternion parameters $\in (\mathbb{R}^{3 \times 1}, \mathbb{R})$  
$a$ Quaternion rotation axis $\in \mathbb{R}^{3 \times 1}$  
$F$ Force $\in \mathbb{R}^{3 \times 1}$  
$h$ Angular momentum $\in \mathbb{R}^{3 \times 3}$  
$M$ Moment $\in \mathbb{R}^{3 \times 1}$  
$q$ Quaternion $\in \mathbb{R}^{4 \times 1}$  
$r$ Position vector $\in \mathbb{R}^{3 \times 1}$  
$u$ Control outputs  
$v$ Linear velocity $\in \mathbb{R}^{3 \times 1}$  
$x$ State variables
Chapter 1

Introduction

This chapter provides the background and motivation for the research, and describes the related literature in the field of hybrid UAVs. The hybrid UAV configurations are evaluated for their benefits and drawbacks, and one of them is selected based on suitability for the wetland inspection mission. This chapter also discusses the research objectives and provides an overview of how the overall thesis document is structured.

1.1 Background and Motivation

The recent growth in the development of unmanned aerial vehicles (UAVs), especially of multirotors, has led to their expanded uses in a wide range of scenarios, such as wildfire monitoring [1], coastal line assessment [2], precision agriculture [3], and wetland inspection [4]. While multirotors are capable of precise maneuvering and vertical take-off and landing (VTOL) without requiring a proper runway, they suffer from some severe shortcomings, such as having low flight speed, short range, and limited endurance. These disadvantages hinder the use of multirotors in applications that require long range and large payload-carrying capabilities.

One of such applications, which is the focus of this thesis, is the wetland algal bloom inspection mission in partnership with scientists of Queen’s University Biology Department. The mission requires the UAV to have an endurance of 30 minutes, a range of 20 km, and be able to perform water take-off and landing to conduct water sampling. To combine the benefits of both multirotors and fixed-wing UAVs, hybrid UAV configurations such as a tail-sitter, a tilt-rotor, and a tilt-wing are studied and compared in Section 1.2.

Hybrid UAVs combine the characteristics of multirotors and fixed-wing UAVs, and operate in two distinct flight regimes of hovering and forward flight. Multirotors are inherently unstable and require active attitude stabilization. The control of multirotors near the hover condition or at small Euler angles has been well established using linearized controllers [5, 6]. Fixed-wing UAVs, on the other hand, are often designed to be statically and dynamically stable in order to be manually piloted. Feedback control is commonly used to control altitude, flight
speed, pitch, and roll angles of fixed wing UAVs [7]. These controllers are derived from equilibrium conditions through small angle approximations and small disturbances theories [7]. Transition dynamics of hybrid UAVs, however, are highly nonlinear, and involve additional complexity in modelling the aerodynamics at low airspeed especially in the stall regime [8]. A transition control strategy is required to switch between the two distinct regimes of multirotor hovering flight and fixed-wing forward flight, while keeping the UAV stable throughout the transition.

1.2 Related Work

Previous work on VTOL-capable fixed wing UAVs mainly focused on three configurations: tail-sitters, tilt-rotors, and tilt-wings. This section describes the design variations of each configuration, the controllers and the transition strategies employed, as well as their benefits and drawbacks.

Tail-Sitter UAVs

A tail-sitter, also known as a prop-hanger, sits on its tail upon take-off and landing, and tilts forward to achieve horizontal flight, usually with the help of control surfaces such as elevators or elevons (a combination of elevator and aileron).

![Variations in Tail-Sitter Design](image)

The main benefit of a tail-sitter is its mechanical simplicity as it requires no additional actuators other than conventional fixed-wing aircraft control surfaces, which reduces complexity, susceptibility to malfunctions, and saves weight [12, 13, 14]. Tail-sitter UAVs can be heavier than fixed-wing UAVs due to the additional structural reinforcement required, as the transition between hover and level flight involves pull up maneuvers that could exert high G-forces on the aircraft structures [14].

Despite the simplicity of tail-sitters, they are susceptible to wind disturbances due to their large vertical wing areas [12], which make precise hovering challenging [15]. The transition of tail-sitter UAVs between hover and forward flight also shows large variations in altitude...
The inability to hover and hold altitude makes tail-sitter UAVs unsuitable for wetland inspection as the UAV is required to precisely land on water to collect samples. Additionally, having the aircraft fuselage pitch between $0^\circ$ and $90^\circ$ requires cameras and sensors to be gimbaled to adapt to such changes in attitude, or their usefulness will be limited to either during hovering or during forward flight.

During hover and transition where the airflow over the wing is limited, the control surfaces rely on the propellers slipstreams to generate control forces [9, 10]. During slower transition scenarios, ailerons are often found to be saturated in counteracting the motor torque [12, 13, 15]. This limited airflow over the lifting surfaces also results in divergent behavior in the yaw axis of the aircraft, as the moment created by the gravity vector not being parallel to the aircraft’s centerline is not sufficiently counteracted by the limited rudder effectiveness [17]. As such, tail-sitter UAVs typically have larger control surfaces than fixed-wing UAVs to increase the effectiveness of those control surfaces [9].

In addition, the high center of gravity (C.G.) of tail-sitter UAVs during the take-off and landing contributes to the lack of ground stability. A solution proposed by Stone involves designing a canard tail-sitter to allow the C.G. to be placed further behind the aircraft [14]. The sizing of a tail-sitter involves trade-offs between having sufficient ground footprint to prevent tip-over and having sufficient lateral stability margin during forward flight [14].

Most tail-sitters presented in the literature use a single motor on a tractor-type aircraft as shown in Figure 1.1(a) [9, 12, 17]. Other variations include using multiple motors such as a pair of coaxial counter-rotating propellers, but this could complicate the aerodynamic model during transition [16]. Stone’s T-wing tail-sitter, as shown in Figure 1.1(b), uses two motors to counteract the gyroscopic effect, but it was found that gain scheduling is necessary for yaw control to account for the effects of differential thrust [10]. The use of multiple motors helps with improving the attitude control of tail-sitters during hover, as observed in the quadrotor tail-sitter design without control surfaces shown in Figure 1.1(c) [11]. Quadshot is a similar quadrotor tail-sitter design proposed by Sinha et al. with elevon controls; however, they noted that elevon reversal could happen during fast, vertical descent in hovering mode [18].

Most tail-sitter literature focused on attitude control during hover with little emphasis on position control. Green used the pitch and yaw errors as feedback for a PD controller to deflect the rudder and elevator, while allowing the aircraft to roll freely under the effect of the motor torque [9]. Escareno simplified the model of a tail-sitter UAV by separating the lateral, longitudinal, and axial dynamics into a chain of four integrators in cascade, and used a saturation-based control scheme to stabilize the vehicle in hover [19]. Quaternions were also used to represent aircraft attitude to prevent gimbal lock when the aircraft pitches up at $90^\circ$ when hovering [9], and to calculate quaternion error for feedback control [17]. Matsumoto et al. presented a control strategy that resolves the attitude errors into tilt-twist angle to increase robustness against larger attitude errors, as using quaternions for feedback in hover only works well with small attitude errors [12].
The transition from hover to forward-flight for a tail-sitter involves accelerating the UAV in the upward direction to gain sufficient airspeed before pitching the nose down to achieve horizontal flight [20]. Alternatively, Stone proposed stalling the tail-sitter and allowing it to tumble to gain airspeed while transitioning into forward flight, but he recognized the potential danger in such a maneuver [14].

To account for modelling error and susceptibility of tail-sitter UAVs to external disturbances, an additional integral term was used in calculating the desired acceleration based on the sum of position errors [13]. Alternatively, an adaptive controller such as model reference adaptive control (MRAC) was used to account for the non-linear dynamics during the transition, and it was found to produce better performance than a PID controller [17]. Johnson et al., on the other hand, implemented “a single hidden-layer perceptron neural network (NN) ... to model the error between the [linearized model around hover condition] and the actual nonlinear aircraft system” and then used the trained NN for real-time adaptation [15].

Tilt-Wing UAVs

Tilt-wing aircraft have been actively studied in the past, with multiple tilt-wing aircraft such as VZ-2, Vertol, and GL-10 being designed and studied by Boeing and NASA. These tilt-wing aircraft have a tiltable main wing with two or more motors mounted to it, as shown in Figure 1.2(a) [21]. The QTW, shown in Figure 1.2(b), acts like a quadrotor in hover and transitions into a tandem wing aircraft in forward flight [22]. Oner et al. proposed a similar design without any control surface, but the authors only demonstrated flight results in hover using a linear-quadratic regulator (LQR) position controller without utilizing the wing-tilting mechanism [23]. The GL-10, on the other hand, includes a tiltable horizontal stabilizer with two motors mounted on it to provide pitch control [24]. Tilt-wing aircraft typically have a higher disc-loading than tilt-rotor aircraft; they are less efficient during hover but are more efficient in cruise [22].
Most tilt-wing aircraft have propellers with large diameters to ensure the propeller slipstream over the wing will prevent the wing from stalling during descent and transition [8, 26]. Wing stalling can also be avoided by having high-lift devices, increased leading edge radii, and large wing chords [8, 21, 26].

Similar to tail-sitters, tilt-wings are susceptible to wind disturbances during take-off and landing [27]. In addition, tilting the entire wing with the rotors on a tilt-wing UAV requires a large amount of torque. The need for housing a more complex tilting mechanism limits the payload-carrying capacity of a tilt-wing UAV [27].

**Tilt-Rotor UAVs**

Tilt-rotor UAVs use actuators to tilt the propulsion system (hence the thrust vector), usually along the longitudinal axis, to allow the UAVs to take-off vertically and subsequently transition into the forward flight mode. Tilt-rotor UAVs enjoy the benefits of maneuverability, hover, and VTOL capability of multirotor UAVs, as well as the endurance, speed, and payload-carrying capacity of fixed-wing UAVs. In contrast to tail-sitters, tilt-rotor UAVs are more capable of maintaining constant or little change in altitude during transition [16, 28].

The most common tilt-rotor UAV has four actutable rotors as shown in Figure 1.3(a) [27]. Wang and Cai use “free wings” as additional actuators to redirect airflow behind the propellers for pitch control as shown in Figure 1.3(b) [16]. Other variation includes a flying wing with two actutable rotors and a pair of fixed coaxial counter-rotating propellers acting as the central lifting fan as shown in Figure 1.3(c) [29]. Rudin et al. proposed a fixed-wing conventional aircraft which uses a pair of tilt-rotors on the wing and a tail-rotor that tilts sideways to act like a tricopter in hover, as shown in Figure 1.3(d) [30].

![Figure 1.3: Variations in Tilt-Rotor UAV Design](image)

Despite the benefits of tilt-rotor UAVs, the actuators required for the tilting mechanism add complexity and weight to the UAV. Other design disadvantage includes the need to account for the C.G. location in both multirotor and fixed-wing configurations, to ensure that the C.G. placement also meets longitudinal static stability while being close to the geometric center of the motors. Modelling of the tilt-rotor dynamics requires consideration of the aerodynamic forces and moments induced by the propellers and lifting surfaces [16, 29], the effectiveness of the propeller slipstream [29], the gyroscopic and adverse reactionary moments induced by the
tilting mechanism [16], as well as the aerodynamics of the wing at post-stall angles of attack.

Several methods were employed to model the flight dynamics. These include wind tunnel tests to help identify the transition corridor [31], sophisticated Computational Fluid Dynamics (CFD) modelling to account for interactions between the wing and propellers [32], system identification and flight modelling through test flights and an adaptive neural network [33], as well as using built-up methods to address the various contributions of forces and moments [16, 34]. Due to difficulties in modelling the aerodynamics of the UAV at low airspeeds, Reference [28] assumes that the aerodynamic forces and moments during hover mode are negligible until the UAV gains sufficient airspeed in forward flight.

Linear controllers are commonly employed for each flight regime (hover, transition, and forward flight) of the tilt-rotor UAV. Reference [35] used PID controllers to provide feedback control on different sets of variables for each flight regime. Similarly, Wang and Cai proposed the use of observer-based PD controllers with feedforward terms for attitude and position control [16]. Dynamic inversion (DI) was also employed in Reference [36] in the outer control loop to determine the outputs of the virtual actuators by inverting the dynamic model of the UAV and using the desired transition trajectory as a reference model. However, dynamic inversion is sensitive to modelling errors of the system and to external disturbances [36, 37]. Alternatives to DI include using an extended-state observer to obtain adaptive compensation through a feedback mechanism on the disturbance variables [36], using NN to adapt to model inversion error [38], and using incremental nonlinear dynamic inversion (INDI) to obtain a linearized system that is insensitive to model uncertainties [37, 39].

During the transition from hover to forward flight, the UAV has to achieve an airspeed that allows the wing surfaces to generate sufficient lift. The most common transition strategy between hover and forward flight involves using a switching algorithm, in which different controllers are activated for different flight modes. The hover controller is initially used to track a desired horizontal speed, and once the desired speed is achieved, the rotors are tilted to a pre-defined angle while activating the transition controller which continues to accelerate the UAV above its stall speed [30, 35]. While the authors in Reference [35] successfully demonstrated stable transition in simulation and in experiment, the chosen combinations of flight conditions and rotor tilt angles to activate the controller switching seem arbitrary and are likely motivated by experiments and test results. A similar switching logic was used by Wang and Cai; the four motors with variable pitch propellers are used in controlling the attitude of the tilt-rotor UAV, and after achieving a pre-defined tilt-angle of 45° of the rotors, the pitch dynamics are controlled by the free-wings attached behind the motors [16].

Another technique to deal with the control of the two different flight modes involves weighing the outputs of the hover controller and the forward flight controller depending on the rotor tilt angle [30]. The gain-scheduling technique is also commonly used to adjust the difference in control gains across the two distinct flight modes. The dynamics of the tilt-rotor UAV are linearized into state-space representation, where the controller gains are determined
through pole placement [40] and the Linear Quadratic Regulator (LQR) method [41].

An important criterion in transition is the variation of altitude throughout transition. The ability to maintain altitude is necessary to ensure sufficient vertical clearance from ground obstacles, and to eliminate the need for additional climb or descent maneuvers after transition. The simulation and experimental results presented in References [35] and [40] showed large change in altitude during transition. Reference [28], on the other hand, proposed a continuous altitude controller based on nested saturation functions that does not involve controller switching. However, in their paper, the total thrust produced by the motor is set constant, which is often not the case given that the thrust required in forward flight is only required to overcome drag and therefore should be less than the thrust required for hovering.

**Chosen Hybrid VTOL UAV Configuration**

In this thesis, the tilt-rotor configuration is chosen for the UAV prototype. This configuration is more controllable and less susceptible to disturbances than a tail-sitter UAV, while being less mechanically complex than a tilt-wing UAV. The design of this tilt-rotor UAV is simplified by combining a quadrotor frame with a tailless aircraft configuration, and by adding actuators to rotate the thrust vector with the quadrotor frame throughout transition. In such a configuration, the wing-body, which houses the payloads, remains above water while the quadrotor frame provides buoyancy for the purpose of water take-off and landing during the wetland inspection mission.

**1.3 Research Objectives and Goals**

The high-level objective of this research is to design and control a UAV that is capable of vertical take-off and landing (VTOL), hovering and performing forward flight for the purpose of wetland algal bloom monitoring and inspection, or for other similar applications that require long flight endurance and high payload-carrying capability.

A novel VTOL tilt-rotor UAV design named AmphiQuad, which combines a quadrotor frame and a flying-wing body capable of water flotation, is proposed. The design, analysis, and selection of components of the UAV are presented in detailed in Chapter 2. The proposed design of the UAV serves to demonstrate the potentials of extending the endurance of existing multicopter frames by mounting a lifting surface with a tilt-rotor interface, and by implementing a custom controller on the open-source Pixhawk autopilot board. The dynamics of the AmphiQuad UAV are presented in Chapter 3; they are constructed by summing up the forces and moments contributed by the motors, aerodynamics of the lifting surfaces, gravity, and gyroscopic moments through a built-up method.

In this thesis, a transition algorithm that uses a gain-scheduled altitude controller and a quaternion-based attitude tracking controller is presented, with the objective of ensuring a stable transition with minimal change in altitude. The altitude controller is developed using
a linearized longitudinal dynamic model of the UAV, with the gains scheduled with the pitch angle of the quadrotor-body frame. These gains are obtained through pole-placement or the linear quadratic regulator (LQR) method. An equilibrium speed update equation is incorporated into the altitude controller to account for possible deviation in the equilibrium speed upon which the linearized model is constructed. The attitude tracking controller is quaternion-based to prevent singularities when the desired pitch angle of the quadrotor-body follows a transition trajectory from 0° (hover) to −90° (forward flight). The altitude and attitude controllers are presented in Chapter 4.

In addition to proposing altitude and attitude controllers, this research also involves studying existing controllers and transition strategies available through the efforts of the open-source ArduPilot community in developing the autopilot firmware. These are discussed in Chapter 5. These controllers are evaluated through simulations in MATLAB Simulink in Chapter 6, for their capability of transitioning into forward flight while maintaining attitude and altitude. The proposed altitude controller is implemented on an open-source autopilot board, and its performance is validated experimentally through flight tests. The experimental results are presented in Chapter 7.
Chapter 2

Design of the AmphiQuad UAV

This chapter first discusses the tools used in the design and analysis process of the UAV. This is followed by the component-level details on the tilt-rotor UAV design, which include the wing design, the quadrotor frame design, the tilt-rotor interface design, propulsion selection and analysis, avionics selection, as well as the construction techniques and material selection. This chapter also presents mass breakdown and the performance analysis of the UAV.

2.1 Design Tools

Multiple design tools were employed to analyze the performance of the UAV. These include MotoCalc for propulsion analysis, as well as XFOIL and AVL for aerodynamic analysis. This thesis does not intend to develop a comprehensive aerodynamic model of the UAV, hence the use of CFD was avoided as it is computationally expensive and time consuming. As XFOIL and AVL use a command line interface and are capable of exporting analysis results into a formatted-text file, MATLAB was used to fully automate and integrate the analysis procedures.

MotoCalc

MotoCalc is a commercial software that contains a database of motors with their technical specifications, such as motor constant, no-load current, and motor resistance. In this thesis, MotoCalc was used for propeller selection and performance analysis of the UAV. Based on the user-defined battery parameters and propeller dimension, the dynamic thrust and power consumption of the motor-propeller combination were estimated as a function of throttle percentage and airspeed [42]. Due to lack of equipment such as load cells and wind tunnel, only the static thrust of the motor was validated using a test stand shown in Figure 2.1.
XFOIL

XFOIL is a tool developed by the Massachusetts Institute of Technology (MIT). It takes in a 2D airfoil coordinates file, and calculates the lift, drag and pitching moment coefficients of the airfoil as a function of angle of attack at a given Reynolds number. This estimation is done using “high-order panel methods with ... fully-coupled viscous/inviscid interaction method” [43, 44]. XFOIL also contains an airfoil geometry design component that allows airfoil customization through changing the leading edge radius, thickness, and camber, as well as interpolating between two airfoils. In this thesis, XFOIL was used to analyze and select 2D airfoils for the main wing and the vertical tails, as well as to provide the 2D drag polars of the airfoils for AVL to estimate the 3D drag coefficient of the UAV.

AVL

AVL is another tool developed by MIT to analyze the aerodynamic and stability of a 3D aircraft. It takes in the definition of the lifting surfaces in the form of an .avl file that contains geometry, position, sections of the lifting surfaces, as well as the airfoil used and the presence of control surfaces on each surface section. More information on the formatting of the .avl file can be found on Reference [45]. The .avl file defining the geometry of the AmphiQuad UAV can be found in Appendix A.1.

In this thesis, AVL was used to calculate the 3D lift, induced drag, pitching moment coefficients of the UAV based on a defined level flight condition. This subsequently provides an estimation of the range and endurance performance of the UAV as presented in Section 2.10. These calculations are based on an inviscid incompressible flow model using Vortex Lattice Method (VLM) [45]. While AVL provides a good estimation of the 3D lift coefficient, it cannot provide an accurate estimation of drag. As the wind tunnel testing and CFD analysis are not within the scope of the research, a simplified drag polar of drag coefficients versus lift coefficients based on the results from XFOIL was included for AVL to estimate the 3D drag coefficients of the UAV through quadratic interpolation between the data points.

MATLAB

As the analysis on AVL and XFOIL requires definition of the flight conditions such as angle of attack, Reynolds number, center of gravity, and etc., repeated runs at different flight conditions and aggregating the results can be tedious and time-consuming. As both software have a command-line interface, the analysis on XFOIL and AVL was fully automated through MATLAB by launching the executable and sending the keystrokes required for performing the analysis. The automation includes creating the input files required for the analysis, inputting user options and flight conditions for the analysis, retrieving the output files, and plotting the results to help visualization. The workflow between XFOIL, AVL, and MATLAB is illustrated in Figure 2.2.
Figure 2.2: Illustration of the Analysis Workflow Between MATLAB, AVL, and XFOIL

2.2 Airfoil and Wing Design

2.2.1 Configuration Selection

The fixed-wing configuration of the AmphiQuad UAV was chosen to be a flying wing, which simplifies construction and makes the conversion of any multirotor frame into a hybrid UAV easy by simply adding a wing. A flying-wing configuration is also less mechanically complex and can be built lighter than conventional aircraft configuration due to the absence of an empennage and the reinforcement required to support such structure.

2.2.2 Airfoil Selection

With a tailless configuration, reflex airfoils were chosen to produce lift and zero pitching moment at positive angle of attack. Geometric and aerodynamic twists were not considered as they increase construction complexity. Several airfoils from the UIUC database (Reference [46] and [47]) were considered and analyzed using XFOIL to obtain the lift, drag, and pitching moment coefficients at a range of angles of attack. Based on the lift-to-drag ratio and maximum lift coefficient at a Reynolds number of 600,000, the choice of airfoils was narrowed down to: Eppler 325, Eppler 342, HS 520, and a custom airfoil chu3 which interpolates between the Eppler airfoils to improve the lift-to-drag and pitching moment curve. The comparisons between the airfoils are shown in Figure 2.3.

The custom airfoil chu3, shown in Figure 2.4, was selected for a more positive $C_m$ at zero angle of attack than the Eppler 342 and HS 520 airfoils. This ensures positive pitch stiffness and that the static equilibrium occurs at positive $\alpha$ [7]. The custom airfoil also has a higher 2D $C_l$ than the Eppler 325 airfoil and slightly lower $C_D$ than the Eppler 342 airfoil at lower angles of attack. A higher lift coefficient allows the UAV to cruise at a lower speed where the dynamic thrust is higher, and is more power-efficient with the same fixed-pitch propellers that
are used for hovering. A lower drag coefficient, on the other hand, enables longer flight range and endurance of the UAV.

![Graphs of aerodynamic coefficients vs angle of attack](image)

**Figure 2.3:** Comparing Performance of Different Airfoils Using XFOIL

**Figure 2.4:** Geometry of the Custom Airfoil Used in the Main Wing of AmphiQuad UAV

### 2.2.3 Planform Design

The criteria considered in sizing the planform of the AmphiQuad UAV include the number of taper sections, wingspan, aspect ratio, taper ratio, and sweep angle.

Additional constraints were considered to improve the ease of construction and transportation of the UAV. These include a maximum wingspan of 2 m, a maximum dimension of 1 m for all permanently interfaced parts, and a total UAV takeoff mass of less than 10 kg. Any
individual part that will be constructed using laser-cutter must fit within a bounding box of 0.3 m by 0.6 m, which is the size limit of the laser-cutter.

The wing was chosen to be tapered to closely model the elliptical lift distribution for planform efficiency, where a taper ratio of 0.4 is generally recommended [48]. However, for the ease of construction, the number of taper sections was restricted to two: one untapered center wing section that houses the payloads and batteries, and one tapered outer wing section which is sweptback to improve lateral and directional static stability [49]. The sweep angle was determined to place the neutral point at 35% of the root chord for longitudinal static stability, while preventing interference with the propellers.

The sizing of the above-mentioned parameters was determined by an optimization script to minimize the error between the local lift distribution $\bar{C}_L(y)$ generated by the wing and the ideal elliptical local lift distribution $\bar{C}_{L,\text{ideal}}(y)$. The optimization function is,

$$ J = \int_{y=0}^{b/2} \left( c(y) \bar{C}_L(y) - c(y) \bar{C}_{L,\text{ideal}}(y) \right)^2 dy $$

where $\bar{C}_{L,\text{ideal}}(y) = \bar{C}_L(y = 0) \sqrt{1 - \frac{y^2}{(b/2)^2}}$

where the local lift coefficients $\bar{C}_L(y)$ are obtained from AVL, and are dependent on the planform design parameters $b$, $c_{\text{root}}$, $c_{\text{tip}}$, $b_{\text{outer}}$, $b_{\text{center}}$, and $S_{\text{LE}}$ as illustrated in Figure 2.5. The optimization was subjected to several design constraints as described in Table 2.1.

<table>
<thead>
<tr>
<th>Constraints</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b \leq 2.0 \text{ m}$</td>
<td>Maximum wingspan for ease of transportation and to reduce root bending moment</td>
</tr>
<tr>
<td>$c_{\text{root}} \leq 0.6 \text{ m}$</td>
<td>Maximum chord length at root to ensure that the rib fits within the laser-cutter</td>
</tr>
<tr>
<td>$c_{\text{tip}} \geq 0.15 \text{ m}$</td>
<td>Minimum chord length at tip to ensure prevent tip stall during turns</td>
</tr>
<tr>
<td>$b_{\text{outer}} \leq 0.6 \text{ m}$</td>
<td>Maximum outer wing section semi-span to ensure it fits within laser-cutter</td>
</tr>
<tr>
<td>$b_{\text{center}} \leq 0.85 \text{ m}$</td>
<td>Maximum center wing section span to ensure that it fits the carbon fiber square tubes as the main spar flanges</td>
</tr>
<tr>
<td>$S_{\text{LE}} \geq 0^\circ$</td>
<td>Positive leading edge sweep (sweptback) angle for roll and directional stability [7]</td>
</tr>
</tbody>
</table>

The code that was written for the planform optimization interfaces with the AVL software and uses the MATLAB built-in `fmincon` solver, which is a “[n]onlinear programming solver [for] constrained nonlinear multivariable function” [50]. The sizing of the planform as a result of the optimization is shown in Section 2.11; it has a wingspan of 2 m, a wing area of 0.882 m$^2$, an aspect ratio of 4.5, a leading edge sweptback angle of 23.8$^\circ$ (equivalent to a quarter-chord sweptback angle of 17.9$^\circ$), and a taper ratio of 0.43 at the outer wing section.

Figure 2.6 shows the lift ($C_L$), drag ($C_D$) and pitching moment ($C_m$) coefficients of both the main-wing body and the vertical-tail (see Section 2.3) obtained from the AVL analysis being plotted as a function of angle of attack. As AVL uses an inviscid solver, the range of angle of attack is limited to $-10^\circ$ to $15^\circ$ to prevent obtaining results in the stall regions of the airfoil.

With a flying-wing configuration, the only control surfaces on the UAV are elevons, which
CHAPTER 2. DESIGN OF THE AMPHIQUAD UAV

Optimization Parameters

- Center wing semispan $b_{center}/2$
- Semispan outer section $b_{outer}/2$
- Leading edge sweep angle $S_{LE}$
- Tip chord $c_{tip}$

Optimization Cost Function

Constraint: $b \leq 1m$

Center wing semispan $b_{center}/2$

Figure 2.5: Planform Optimization Parameters and Final Optimization Cost

Figure 2.6: Aerodynamics Coefficients From AVL Analysis Vs. Angle Of Attack

act as both elevators and ailerons. The sizing of these control surfaces was based on the general rule of thumb for model aircraft [49], where the width of each elevon is 35% of the semi-span of the aircraft, with a length of 25% of the chord length at that section.

2.3 Quadrotor Frame Design

The two most common quadrotor configurations are the X-frame and the H-frame, which reflect the shape of the frame and how the motors are mounted on the frame. While both configurations are functionally similar, the structural differences can result in varying torsional strength and will determine how the quadrotor frame is interfaced with the wing-body.

For the AmphiQuad UAV, the wing-body design detailed in Section 2.2 requires a H-frame configuration for the quadrotor frame to minimize interference with the main wing. The H-frame was constructed by joining four carbon fiber (CF) square tubes together, as shown in Figure 2.7. Square tubes were chosen as they are easy to interface with other components of the UAV. This H-frame design allows the attachment of two vertical tails on each side of the frame. These lifting surfaces serve two purposes: to provide directional stability in fixed-wing
configuration and to provide buoyancy when taking-off or landing on water. The vertical tails were sized to have a vertical tail volume coefficient \( V_v \) of 0.057 (>0.05) as a general design rule of thumb to ensure that it provides sufficient directional stability [51].

The airfoil chosen for the vertical tails is one of the symmetrical NACA 4-digit airfoils, with its thickness determined by the amount of water it can displace before reaching a level that will be in contact with the avionics. As the NACA airfoils have an analytical equation for thickness as a function of chord length as shown in Eq. (2.4) [52], the volume of vertical tail was calculated through numerical integration of the area of the 2D airfoil at each span-wise increment of the two vertical tails. NACA0020 was chosen to be the airfoil of the vertical tails; the airfoil has a maximum thickness of \( t = 0.20 c(y) \), where \( c(y) \) is the chord length of the airfoil at the span-wise location along \( y \). The result in Figure 2.8 shows that the vertical tails are capable of supporting the take-off mass of the UAV at 5.45 kg, up to a maximum mass of 7.8 kg. The flotation ability of the UAV was tested in a pond at the Koffler Scientific Reserve (KSR) at Jokers Hill, as shown in Figure 2.9.

\[
\begin{align*}
    z_t(x) &= 5t \left[ 0.2969 \sqrt{\frac{x}{c}} - 0.1260 \left( \frac{x}{c} \right) - 0.3516 \left( \frac{x}{c} \right)^2 + 0.2843 \left( \frac{x}{c} \right)^3 - 0.1015 \left( \frac{x}{c} \right)^4 \right] \\
    A_y(y) &= 2 \int_{x_{\text{start}}}^{x_{\text{end}}} z_t(x) \, dx \\
    V &= 2 \int_{-b_{VT}}^{b_{VT}} A_y(y) \, dy = 4 \int_{0}^{b_{VT}} A_y(y) \, dy
\end{align*}
\]  

The length of the motor arms was chosen to ensure that there will be no interference between the propellers and the wing, as well as between the propellers and the vertical tails throughout the transition from hover to forward flight, as shown in Figure 2.10.

### 2.4 Tilt-Rotor Interface Design

The quadrotor frame, which pitches forward during the transition from hover to forward flight, interfaces with the main wing through a revolute joint and is driven by two servos.
The initial design is shown in Figure 2.11(a), where the servo gear was used to drive a shaft that connected to the carbon fiber (CF) tubes on the vertical tail of the quadrotor frame. However, due to lack of CF tubes sufficiently long to run through the entire center wing section, the shaft had to be in two separate pieces, one on each side of the center wing. This causes the shaft, which holds the weight of half the quadrotor frame, to be cantilevered and only supported against the outermost rib. The deflection in the shaft prevents proper engagement of the gears to achieve a reliable tilting motion. In addition, during ground test, it became evident that the shear strength of the adhesive at the interface between the shaft and the quadrotor frame, as well as the interface between the shaft and the gear, is insufficient for high torque servo-tilting operations.

A substantial design revision was made to overcome the above-mentioned issues, which
involved choosing stronger servos, directly mounting the servos to the rib, and directly mounting the servo attachment to the quadrotor frame. A custom mount that interfaces with the servo attachment (as shown in Figure 2.11(b)) was designed and 3D-printed, with four bolts holding them to the quadrotor frame securely. Ground tests were conducted to repeatedly operate the servos to ensure reliability of this tilting mechanism.

2.5 Wing Sections Interface Design

To ease construction challenge, transportation and storage of the UAV, the 2 m wingspan UAV is divided into three sections: one center wing section and two outer wing sections. These sections are located at where the vertical tails would be in forward flight, as shown in Fig-
These sections are joined together through a CF rod that acts as a wing joint. The rod permanently attaches to the outer wing section through adhesive and was further reinforced to transfer the bending load to the main spar, as shown in Figure 2.12. On the other end, the rod is inserted between the CF square tubes that form the top and bottom flanges of the main spar in the center wing. To prevent rotational movement of the outer wing, an additional rod is used near the leading edge to aid alignment during installation of the wing.

![Figure 2.12: Wing Sections Interface Design](image)

The interface between the center wing and the outer wings was ground tested using a three points bending test, where the UAV was held up on the wingtips to simulate the bending load the spars have to handle in level flight in fixed-wing mode. The spars and the interface between the center and outer wings passed the test successfully.

### 2.6 Propulsion Selection and Analysis

The AmphiQuad UAV is electrically powered for greater ease of setup and operation as compared to using gasoline engines. The motors are electric brushless outrunner motors OMA-5010-810 manufactured by O.S. Engines. These motors were chosen based on the manufacturer specifications on the range of maximum static thrusts that they can produce, which overlaps with the expected maximum take-off weight of the UAV with sufficient thrust margin.

The performance of the motor-propeller combination was analyzed using MotoCalc as described in Section 2.1. Several propeller sizes recommended by the manufacturer were analyzed [53, 54], and their maximum dynamic thrusts are plotted in Figure 2.13 as a function of the airspeed. Fixed-pitch propellers were chosen to avoid the mechanical complexity of variable-pitch propellers, but at the expense of higher power consumption during hover.

The propellers were chosen to be 13"×6.5" based on their high static thrust at hovering and low current draw during forward flight. They are capable of generating a total of 12.8
kg of static thrust with a 4-cell Lithium Polymer (LiPo) battery. With a take-off mass of 5.45 kg, this amount of thrust would provide sufficient margin for stability and control. While a 12”×8” and a 13”×8” propellers can generate thrust up to an airspeed of 30 m/s, the UAV was not designed to fly at this speed, and the high current draw at this airspeed would lead to reduced endurance. The current draw and the dynamic thrust of the chosen motor-propeller combination are shown in Figure 2.14 as a function of airspeed and throttle percentage.

### 2.7 Construction Techniques and Material Selection

As the AmphiQuad UAV serves as a proof-of-concept prototype for a novel tilt-rotor UAV design, the choice of materials was mainly motivated by commercial availability, cost, ease of construction and assembly. The chosen materials are easily shaped into the desired geometry through low-cost manufacturing techniques, such as foam-cutting, laser-cutting, and
3D-printing. Future iterations of the design can be made lighter and more precise with more advanced techniques such as Computer Numerical Control (CNC) and injection moulding.

![Diagram of Spar-and-Ribs Design Technique](image1.png)

**Figure 2.15: Spar-and-Ribs Design Technique**

In order to save weight on non-load bearing members, the main wing uses the spar-ribs technique as shown in Figure 2.15. The ribs containing the shape of the airfoil were fabricated using laser-cutter from 1/16” plywood with weight-saving cut-outs. The spar for the center wing section has a box-shaped design as shown in Figure 2.16(a), with commercially available carbon fiber square tubes forming the top and bottom flange, and 1/16” plywood being the shear web. The space between the top and bottom flanges, as described in Section 2.5, forms the interface between the center wing and the outer wing sections. On the other hand, the spar of the outer wing section is made out of plywood-balsa sandwich, as shown in Figure 2.16(b).

![Diagram of Main Spar Design](image2.png)

**Figure 2.16: Main Spar Design**

In addition to the rendering shown in Figure 2.15, the leading edge of the wing is lined with extruded polystyrene (XPS) foam as shown in Figure 2.17 to maintain the shape of the airfoil around corners. The rear spars are constructed using 1/4” Balsa wood, and they are glued to the side of the airfoil ribs. The elevons are made out of 1/16” plywood ribs glued to a CF rod, with the rod being driven by a servo mounted to the rib as shown in Figure 2.18. These lifting surfaces are covered with thermoplastic material using a hobby iron.
The quadrotor frame is structurally held together by four CF tubes forming a rectangle as described in Section 2.3. These tubes were originally connected together using right-angled 3D printed parts as shown in Figure 2.19. However, the poor torsional stiffness of the frame as evident by the poor yaw control performance during flight tests necessitated a redesign where the square tubes are joined permanently with CF strips, as shown in Figure 2.20. The connections between the tubes were further reinforced by mounting the commercially available off-the-shelf CF landing gears across both tubes at the joint.

The CF tubes of the quadrotor frame are embedded within the vertical tails that provide buoyancy during take-off and landing on water. The vertical tails are made out of solid foam as shown in Figure 2.21, and shaped using a foam-cutter available in the aerospace design lab at UTIAS. Due to the tapered nature of the vertical tails, each vertical tail is constructed in
two separate pieces, and they are joined together using foam-safe adhesives and thermoplastic covering.

The motor mounts were custom-designed and 3D-printed to fit the selected motors and the CF tubes they are mounted on. The motor mount design shown in Figure 2.22 went through several design iterations to improve the bending strength and to reduce structural flex, as well as to improve the ease of 3D printing.

2.8 Avionics Selection

For guidance, navigation, and control of the AmphiQuad UAV, the Pixhawk autopilot board was chosen. It is very commonly used by the recreational flying community and by researchers as it is open-source, readily interfaces with other avionics and sensors, and has strong support from the developer community. The Pixhawk autopilot board runs the open-source Ardupilot firmware which will be discussed in Chapter 5.

Several other avionics and sensors were chosen to complement the Pixhawk autopilot board. These include a power module to monitor battery voltage and current, an external GPS and compass module for position and heading information, a telemetry kit to transmit flight data to the ground station in real-time for flight monitoring, a receiver paired to a Futaba t8j transmitter for remote piloting the UAV and for switching between flight modes, an airspeed sensor, and an arming switch to prevent the motors from inadvertently spinning up when they are not in use. The airspeed sensor is mounted at the leading edge near the outboard section of the center wing, while the external GPS and compass are mounted with a vertical offset from the wing to minimize electromagnetic interference (EMI) from the battery and high-current wires that can lead to fluctuations in the magnetometer readings.

The autopilot board is connected to the motors and servos to provide pulse-width modulation (PWM) signals to control the rotational speeds of each of the four motors through the Electronic Speed Controllers (ESC), the angles of the tilt-rotor servos, and the angles of each
control surface. The wiring connections are as shown in Figure 2.23 along with the corresponding parameters in the autopilot firmware.

![Wiring Connections from Pixhawk Autopilot Board](image)

**Figure 2.23:** Wiring Connections from Pixhawk Autopilot Board

### 2.9 Mass Breakdown

The AmphiQuad UAV was initially designed with a target take-off mass of 4.5 kg, with a payload mass of 1.5 kg. A high-level allocation of the mass budget is shown in Table 2.2.

<table>
<thead>
<tr>
<th>Components</th>
<th>Mass (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airframe</td>
<td>1.10</td>
</tr>
<tr>
<td>Propulsion</td>
<td>0.75</td>
</tr>
<tr>
<td>Battery</td>
<td>1.00</td>
</tr>
<tr>
<td>Avionics</td>
<td>0.15</td>
</tr>
</tbody>
</table>

However, as the construction of the UAV progressed, it became evident that this mass budget was too optimistic for a prototype, and hence the UAV will not be equipped with any payload to meet the original take-off mass target. The revised total mass is 5.45 kg, and the mass breakdown is summarized in Table 2.3.

Future weight-saving measures on the AmphiQuad UAV can be made possible by eliminating the solid foam vertical tails and replacing them with air-tight lightweight carbon fiber shells, as well as constructing the wing sections using carbon fiber layups. Due to time constraints and the lack of carbon fiber manufacturing equipment, these techniques were not explored.
Table 2.3: Revised Mass Breakdown for AmphiQuad UAV

<table>
<thead>
<tr>
<th>Components</th>
<th>Mounted on</th>
<th>Subsystem</th>
<th>Mass (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main wing spars</td>
<td>Main Wing</td>
<td>Airframe</td>
<td>0.25</td>
</tr>
<tr>
<td>Main wing ribs</td>
<td>Main Wing</td>
<td>Airframe</td>
<td>0.18</td>
</tr>
<tr>
<td>Main wing covering</td>
<td>Main Wing</td>
<td>Airframe</td>
<td>0.05</td>
</tr>
<tr>
<td>Outer wing spars (x2)</td>
<td>Outer Wing</td>
<td>Airframe</td>
<td>0.18</td>
</tr>
<tr>
<td>Outer wing ribs (x2)</td>
<td>Outer Wing</td>
<td>Airframe</td>
<td>0.17</td>
</tr>
<tr>
<td>Outer wing covering (x2)</td>
<td>Outer Wing</td>
<td>Airframe</td>
<td>0.06</td>
</tr>
<tr>
<td>Outer wing interface</td>
<td>Outer Wing</td>
<td>Airframe</td>
<td>0.08</td>
</tr>
<tr>
<td>Elevon (x2)</td>
<td>Outer wing</td>
<td>Airframe</td>
<td>0.03</td>
</tr>
<tr>
<td>Quadrotor frame</td>
<td>Quadrotor</td>
<td>Airframe</td>
<td>0.37</td>
</tr>
<tr>
<td>Vertical tail (x2)</td>
<td>Quadrotor</td>
<td>Airframe</td>
<td>0.39</td>
</tr>
<tr>
<td>Motor mount (x4)</td>
<td>Quadrotor</td>
<td>Airframe</td>
<td>0.12</td>
</tr>
<tr>
<td>Landing gear (x4)</td>
<td>Quadrotor</td>
<td>Airframe</td>
<td>0.18</td>
</tr>
<tr>
<td>Tilt-rotor interface (x2)</td>
<td>Quadrotor</td>
<td>Airframe</td>
<td>0.11</td>
</tr>
<tr>
<td>Motor (x4)</td>
<td>Quadrotor</td>
<td>Propulsion</td>
<td>0.94</td>
</tr>
<tr>
<td>Propeller (x4)</td>
<td>Quadrotor</td>
<td>Propulsion</td>
<td>0.12</td>
</tr>
<tr>
<td>Battery (x2)</td>
<td>Main Wing</td>
<td>Propulsion</td>
<td>1.43</td>
</tr>
<tr>
<td>Autopilot board + Avionics</td>
<td>Main Wing</td>
<td>Avionics</td>
<td>0.15</td>
</tr>
<tr>
<td>Wires and connectors (multiple)</td>
<td>Main Wing</td>
<td>Avionics</td>
<td>0.13</td>
</tr>
<tr>
<td>Tilt-rotor servo (x2)</td>
<td>Main Wing</td>
<td>Avionics</td>
<td>0.39</td>
</tr>
<tr>
<td>Elevon servo (x2)</td>
<td>Outer Wing</td>
<td>Avionics</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>Main/Outer Wing</td>
<td>Total</td>
<td>3.22</td>
</tr>
<tr>
<td></td>
<td>Quadrotor</td>
<td>Total</td>
<td>2.23</td>
</tr>
</tbody>
</table>

2.10 Performance Validation

Combining the aerodynamic and the propulsion analysis presented in Section 2.2 and 2.6 respectively, the performance limit, range and endurance of the UAV were calculated.

Assuming the UAV cruises at a small angle of attack such that \( \cos \alpha \approx 1 \) and \( \sin \alpha \approx 0 \), the thrust required is expressed in Eq. (2.6) and plotted in Figure 2.24(a). The \( L/D \) ratio is plotted as a function of \( V \) in Figure 2.24(b). At airspeed \( V \), the thrust required for cruise was calculated based on Eq. (2.6) and the current consumption was estimated. The current consumption as a function of airspeed is shown in Figure 2.24(c); the minimum current draw is 20.3 A which occurs at \( V=11.4 \) m/s. This translates into an endurance of 29.6 minutes in forward flight with 4-Cell Lithium Polymer (LiPo) batteries with a total capacity of 10 Ah. In practice, this endurance may be an overestimation due to the Peukert effect [55]. However, an accurate battery model was not constructed as it was not within the scope of the thesis.

\[
V = \sqrt{\frac{W}{\frac{1}{2} \rho S C_L(\alpha)}} \quad (2.5)
\]

\[
T \approx D = \frac{1}{2} \rho V^2 S C_D(\alpha) = \frac{W}{C_L/C_D} \quad (2.6)
\]

The maximum range of the UAV was calculated by plotting the endurance of the UAV as a function of airspeed, and multiplying the endurance at each data point by the airspeed \( V \). This results in Figure 2.25, with the maximum range of the UAV at 21.5 km when \( V=13.3 \) m/s. The
drag at cruise in forward flight and the maximum thrust available are plotted against airspeed in Figure 2.26 to estimate the maximum airspeed that the UAV can operate at, which is 23 m/s.

The maximum endurance of the UAV in hovering mode was determined by equating the thrust required to the take-off weight \( T_{\text{req}} = W \), and evaluating the current-thrust curve at static condition \( (V=0 \text{ m/s}) \). The estimated total current draw of all motors at hover is to be 125.7 A. With a combined battery capacity of 10 Ah, the UAV can hover for 4.8 minutes. The short endurance in VTOL mode is expected due to the relatively high take-off mass of the UAV and the inefficiency of high-pitch propellers in static condition. Hence, the intended operation of the UAV is to perform vertical take-off and quickly transition into forward flight mode after gaining sufficient altitude. Every 1 minute spent in hovering flight will cost approximately 6 minutes of endurance in forward flight.
2.11 Summary of AmphiQuad UAV Design

- Wingspan of 2 m and area of 0.882 m² with two tapered sections
- Elevon (elevator + aileron) for fixed-wing roll/pitch control
- Servo mechanism for tilting the motors forward and help keeping the wing levelled (in pair to provide more torque)
- Vertical tail made out of low-density foam to provide buoyancy for water takeoff and landing
- Motors capable of producing 12.8 kg of thrust in total
- Four carbon fiber square tubes form main support of quadrotor frame
- Batteries mounted in front of main spar
- Autopilot mounted behind main spar
- Carbon fiber landing gear to support takeoff from land
- Carbon-fiber rods for interface between outer wings and center wing
- Spar-rib construction technique with the help of precision lasercutter

The detailed design of the UAV was done through SolidWorks Computer-Aided Design (CAD) software, which allows us to dimension and define the mass properties of the parts and components of the AmphiQuad UAV. Figure 2.27 shows a labelled CAD model that summarizes the major components and their layout on the UAV. In summary, a tilt-rotor configuration was chosen for better hovering performance than a tail-sitter, and for being less mechanical complex than a tilt-wing configuration. The design of the tilt-rotor was simplified by combining a quadrotor body and a tailless wing-body, which are interfaced together through a servo-tilting mechanism. This allows a pair of vertical tails to be mounted to the quadrotor.
frame, which serves to improve directional stability in fixed-wing configuration, and to pro-
vide buoyancy when taking off and landing on water during the wetland inspection mission.

The important parameters of the UAV are summarized in Table 2.4, and will be used for
the simulation in Chapter 6. An engineering drawing of the UAV is shown in Figure 2.28.

**Table 2.4**: Dimensions and Mass Properties of AmphiQuad UAV

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main wing span, $b_{wing}$</td>
<td>2.0</td>
<td>m</td>
</tr>
<tr>
<td>Main wing area, $S_{wing}$</td>
<td>0.882</td>
<td>$m^2$</td>
</tr>
<tr>
<td>Mean aerodynamic chord, $c_{wing}$</td>
<td>0.473</td>
<td>m</td>
</tr>
<tr>
<td>Vertical tail span, $b_{VT}$</td>
<td>0.5</td>
<td>m</td>
</tr>
<tr>
<td>Vertical tail area, $S_{VT}$</td>
<td>0.144</td>
<td>$m^2$</td>
</tr>
<tr>
<td>Vertical tail mean chord, $c_{VT}$</td>
<td>0.288</td>
<td>m</td>
</tr>
<tr>
<td>Main wing mass, $m_{wing}$</td>
<td>3.22</td>
<td>kg</td>
</tr>
<tr>
<td>Main wing inertia, $I_{wing</td>
<td>x,y,z}$</td>
<td>$[0.42, 0.04, 0.40]^T$</td>
</tr>
<tr>
<td>Main wing C.G. in $\vec{F}<em>B$, $r</em>{CG}$</td>
<td>$[0.259, 0.0, 0.0]^T$</td>
<td>m</td>
</tr>
<tr>
<td>Vertical tail mass, $m_{VT}$</td>
<td>2.23</td>
<td>kg</td>
</tr>
<tr>
<td>Vertical tail inertia, $I_{VT</td>
<td>x,y,z}$</td>
<td>$[0.16, 0.18, 0.32]^T$</td>
</tr>
<tr>
<td>Vertical tail C.G. in $\vec{F}<em>Q$, $r</em>{CG}$</td>
<td>$[0.0, 0.0, 0.173]^T$</td>
<td>m</td>
</tr>
</tbody>
</table>

**Figure 2.28**: Engineering Drawing of AmphiQuad UAV (In Forward Flight Mode)

Note that the engineering drawing was originally created for an A3 paper, and were scaled by
a factor of 0.3 to be included in this thesis.
Chapter 3

Dynamics of the AmphiQuad UAV

This chapter first describes the reference frames used throughout the thesis, then presents the equation of motions as well as the calculations for the center of gravity (C.G.) and combined moment of inertia of the UAV. This is followed by the derivation of the dynamics of UAV using a built-up method that combines the forces and moments generated by the aerodynamics of the lifting surfaces, the propulsion performance of the motor-propeller combination, gravity, and the gyroscopic moments from the propellers and the tilt-rotor.

3.1 Reference Frames

Note: the origin of the quadrotor-body and wing-body frame of reference is at the center of mass of the UAV. They are isolated in this figure to ensure clarity of the illustration.

Figure 3.1: Illustration of the Reference Frames and Axes
Four reference frames are used throughout this section, namely the inertial frame $\bar{F}_E$, the wing-body frame $\bar{F}_B$, the quadrotor-body frame $\bar{F}_Q$, and the wind frame $\bar{F}_W$. Similar to most aircraft dynamics literature [7], the inertial reference frame $\bar{F}_E$ is defined such that the $x$-axis points North, the $y$-axis points East, and the $z$-axis points down with altitude, $h = -z$. For the quadrotor-body frame $\bar{F}_Q$ in hover, the $x$-axis represents the UAV heading, $y$-axis points towards the right wing, and $z$-axis points downwards. In forward flight, the $x$-axis of $\bar{F}_Q$ points downwards and the $z$-axis points to the opposite of the UAV heading. The rotation matrices relating those reference frames are follow,

$$
\begin{align*}
\bar{F}_B & = C_1(\phi)C_2(\theta)C_3(\psi) \bar{F}_E = C_{BE} \bar{F}_E \\
\bar{F}_B & = C_2(\gamma) \bar{F}_Q = C_{BQ} \bar{F}_Q \\
\bar{F}_B & = C_2(\alpha) \bar{F}_W = C_{BW} \bar{F}_W
\end{align*}
$$

where:

$$
C_1(\cdot) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\cdot) & \sin(\cdot) \\ 0 & -\sin(\cdot) & \cos(\cdot) \end{bmatrix}
$$

$$
C_2(\cdot) = \begin{bmatrix} \cos(\cdot) & 0 & -\sin(\cdot) \\ 0 & 1 & 0 \\ \sin(\cdot) & 0 & \cos(\cdot) \end{bmatrix}
$$

$$
C_3(\cdot) = \begin{bmatrix} \cos(\cdot) & \sin(\cdot) & 0 \\ -\sin(\cdot) & \cos(\cdot) & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

with $\phi$, $\theta$, and $\psi$ representing the roll, pitch, and yaw angles respectively; $\gamma$ being the tilt-angle of the quadrotor-body ($\gamma = 0^\circ$ for hovering and $\gamma = 90^\circ$ for forward flying); and $\alpha$ representing the angle of attack of the wing-body.

This chapter presents the dynamics of the tilt-rotor UAV in two different reference frames: the wing-body frame $\bar{F}_B$ and the quadrotor frame $\bar{F}_Q$. While they are mathematically equivalent, deriving controllers in both reference frames provides the flexibility of mounting the autopilot board either on the quadrotor frame or within the wing-body. The subscript $B$ used in the equation of motion in Section 3.2 can be replaced by either $B$ or $Q$, depending on whether the wing-body axes or the quadrotor-body axes are used, with the contribution of forces and moments being transformed into the appropriate reference frame as per Eq. (3.1). Note that the quadrotor-body axes $\bar{F}_Q$ are used for the attitude controller and MATLAB simulation that will be presented in the later chapters, as we intend to mount the autopilot board on the quadrotor-body.
3.2 Equations of Motion

This section presents the modelling of flight dynamics for the proposed AmphiQuad UAV design. The origin of the reference frame is located at the combined C.G. of the UAV, calculated in Section 3.3. Following the derivations presented in Reference [7], the dynamics of the UAV are shown in the form of Newton-Euler equations, assuming a quasi-steady state,

\[
\begin{align*}
\mathbf{f}_B &= m(\mathbf{\dot{V}}_B + \omega^\times_B \mathbf{V}_B) \\
\mathbf{G}_B &= \mathbf{\dot{h}}_B + \omega^\times_B \mathbf{h}_B
\end{align*}
\]

which are expanded into,

\[
\begin{align*}
X &= m(\dot{u} + qw - rv) \\
Y &= m(\dot{v} + ru - pw) \\
Z &= m(\dot{w} + pv - qu) \\
L &= I_x \dot{p} - I_{xz}(r + pq) + qr(I_z - I_y) \\
M &= I_y \dot{q} - I_{xz}(r^2 - p^2) + rp(I_x - I_z) \\
N &= I_z \dot{r} - I_{xz}(\dot{p} - qr) + pq(I_y - I_x)
\end{align*}
\]

where \( \mathbf{f}_B = [X, Y, Z] \) are the forces in the body axes, and \( \mathbf{G}_B = [L, M, N] \) are the moments about the \( x, y, z \) body axes respectively. The equations assume geometric symmetry in the \( xz \)-plane, where \( I_{xy} = I_{yz} = 0 \). The terms \( \mathbf{V}_B = [u, v, w] \) are the translational velocities in the body axes, while \( \omega_B = [p, q, r] \) are the body angular rates.

Due to the change in relative position and orientation between the quadrotor-body and wing-body, the combined center of gravity and the moment of inertia of the UAV change throughout the transition. The rates of change in C.G. and inertia are considered to have negligible effects on the dynamics, assuming a slow transition from hover to forward flight.

The forces and moments are contributed by: the propulsion system, gravity, the aerodynamics of the wing-body, and the gyroscopic couples from the propellers and tilt-rotors. The sum of all the forces and moments is then used to calculate the rate of change in angular velocity and the linear acceleration of the tilt-rotor frame. These velocity vectors are integrated to obtain the angular velocity and translational velocity, and then transformed to the inertial frame according to Eq. (3.10) before being integrated again to obtain the position and attitude of the UAV.

\[
\begin{bmatrix}
\phi \\
\dot{\theta} \\
\psi
\end{bmatrix}
= \begin{bmatrix}
1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\
0 & \cos \phi & -\sin \phi \\
0 & \sin \phi \sec \theta & \cos \phi \sec \theta
\end{bmatrix}
\begin{bmatrix}
p \\
q \\
r
\end{bmatrix}
\]

where \([\phi, \theta, \psi]\) are the roll, pitch, yaw Euler angles.
To avoid singularities in Eq. (3.10) when using the quadrotor-body reference frame as the pitch angle approaches $-90^\circ$ in forward flight, quaternions are used to represent the UAV attitude as defined in Eq. (3.11), with the quaternion rates calculated using Eq. (3.12).

$$\begin{align*}
e & = a \sin \frac{\Phi}{2} \\
\eta & = \cos \frac{\Phi}{2} \\
\dot{e} & = -\frac{1}{2} \omega_B^T e + \frac{1}{2} \eta \omega_B \\
\dot{\eta} & = -\frac{1}{2} \omega_B^T \eta
\end{align*}$$ (3.11)

(3.12)

where $a$ is the axis of rotation, and $\Phi$ is the rotation angle along $a$. The quaternion rates are numerically integrated and normalized to satisfy the equation $e^T e + \eta^2 = 1$. The rotation matrix can be constructed from the quaternions according to Reference [56],

$$C_{BE} = \begin{bmatrix} \eta^2 + \varepsilon_1^2 - \varepsilon_2^2 - \varepsilon_3^2 & 2\varepsilon_1\varepsilon_2 + 2\eta\varepsilon_3 & 2\varepsilon_1\varepsilon_3 - 2\eta\varepsilon_2 \\
2\varepsilon_1\varepsilon_2 - 2\eta\varepsilon_3 & \eta^2 - \varepsilon_1^2 + \varepsilon_2^2 - \varepsilon_3^2 & 2\varepsilon_2\varepsilon_3 + 2\eta\varepsilon_1 \\
2\varepsilon_1\varepsilon_3 + 2\eta\varepsilon_2 & 2\varepsilon_2\varepsilon_3 - 2\eta\varepsilon_1 & \eta^2 - \varepsilon_1^2 - \varepsilon_2^2 + \varepsilon_3^2 \end{bmatrix}$$ (3.13)

The velocity in the inertial frame of reference is calculated by applying the rotation matrix on the translational velocities in the body axes, and numerically integrating them to obtain the position of the UAV in the inertial frame,

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{z}_E \end{bmatrix}^T = C_{EB} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^T$$ (3.14)

where $C_{EB} = C_{BE}^T = [C_1(\phi)C_2(\theta)C_3(\psi)]^T$

### 3.3 Center of Gravity (C.G.) and Combined Moment of Inertia

The UAV consists of two bodies, the wing-body and the quadrotor-body, which are connected together through a tilting mechanism. As the quadrotor-body pitches forward during transition from hover to forward flight, the C.G. shifts backward and the combined moment of inertia changes.

The C.G. of the UAV, represented by the terms $h_x$ and $h_z$ as a percentage of the chord length, is calculated as a function of the tilt-rotor angle, $\gamma$ in Eq. (3.15) with Figure 3.2 as an illustration. The mass distribution is assumed to be symmetric in the xz-plane (i.e. $h_y = 0$), and that the C.G. of the wing lies along the chord length. In general, $x_T = 0$, which implies that the C.G. of the quadrotor-body is the geometric center of the four motors.

$$\begin{align*}
h_x & = \frac{1}{m_W + m_T} (m_W h_W + m_T (h_T + \frac{x_T}{\ell} \sin \gamma + -\frac{x_T}{\ell} \cos \gamma)) \\
h_z & = \frac{m_T}{m_W + m_T} (-\frac{x_T}{\ell} \cos \gamma - \frac{x_T}{\ell} \sin \gamma)
\end{align*}$$ (3.15)
Figure 3.2: Calculating Overall Center of Gravity (C.G.) of UAV

The moment of inertia of the UAV is calculated by combining the moment of inertia of the wing-body and quadrotor-body through parallel-axis theorem,

\[
I_B = I_{W,B} + m_W \left[ (r_{B\mid CG_W}) \cdot (r_{B\mid CG_W}) 1_{3\times3} - (r_{B\mid CG_W})(r_{B\mid CG_W})^T \right] + ... \quad (3.16)
\]

\[
I_Q = C_{QB}I_{T,Q}C_{QB}^T + m_T \left[ (r_{B\mid CG_T}) \cdot (r_{B\mid CG_T}) 1_{3\times3} - (r_{B\mid CG_T})(r_{B\mid CG_T})^T \right]
\]

where the moment of inertia of the wing-body \((I_{W,B})\) and quadrotor-body \((I_{T,Q})\) at their respective C.G. are obtained from the CAD model, and the vector from the C.G. of each body to the new combined C.G. is calculated according to Eqs. (3.17) and (3.18). The subscript “W” represents the wing-body, and the subscript “T” represents the tail or quadrotor-body.

\[
r_{B\mid CG_W} = - \begin{bmatrix} h_x \ddot{c} - h_W \ddot{c} \\ 0 \\ h_z \ddot{c} \end{bmatrix} \quad (3.17)
\]

\[
r_{B\mid CG_T} = - \begin{bmatrix} h_x \ddot{c} - (h_T \ddot{c} + z_T \sin \gamma - x_T \cos \gamma) \\ 0 \\ h_z \ddot{c} - (-z_T \cos \gamma - x_T \sin \gamma) \end{bmatrix} \quad (3.18)
\]

Note that the negative sign ahead of the position vector \(r_B\) is added because the definition of the C.G. uses the +z up, +x right convention with respect to the leading edge of the wing, while the reference frames \(\mathbf{\bar{F}}_Q\) and \(\mathbf{\bar{F}}_B\) use the +z down, +x left convention.
3.4 Contribution of Forces and Moments

Propulsion

Figure 3.3 shows the index of each motor on the UAV, which is consistent with the notation used by the ArduPilot firmware [57]. The airspeed perpendicular to the propeller plane is used to calculate the dynamic thrust of the motors as shown in Figure 2.14 in Section 2.6, assuming negligible aerodynamic interactions between the wing-body and vertical tails. The moment contributed by the propeller drag is estimated to be a constant factor, \( k_m \), of the thrust produced [5], due to the lack of equipment to accurately measure this quantity.

\[ F_{\text{motor},Q|c} = \begin{bmatrix} 0 \\ -T_1 - T_2 - T_3 - T_4 \end{bmatrix} \]  

\[ M_{\text{motor},Q|c} = \begin{bmatrix} (T_1 - T_2 + T_3 - T_4) l_{m,x} \\ (T_1 - T_2 + T_3 - T_4) l_{m,y} \\ M_1 + M_2 - M_3 - M_4 \end{bmatrix} \]  

The propulsion dynamics are expressed in the wing-body frame \( \mathbf{F}_B \) about the C.G. of the UAV,

\[ F_{\text{motor},B} = C_{BQ} F_{\text{motor},Q|c} \]  

\[ M_{\text{motor},B} = (r_{B|c} \times F_{\text{motor},B}) + C_{BQ} M_{\text{motor},Q|c} \]  

\[ r_{B|c} = - \begin{bmatrix} h_T \dot{e} + z_T \sin \gamma - x_T \cos \gamma - h_x \dot{e} \\ 0 \\ -z_T \cos \gamma - x_T \sin \gamma - h_z \dot{e} \end{bmatrix} \]
Aerodynamics

The aerodynamic forces and moments are calculated using the non-dimensional terms obtained from analyzing the airfoil and wing geometry in AVL as described in Section 2.2.

As AVL allows the forces and moments to be calculated in the body-axes with the origin being the C.G. of the UAV, this simplifies the aerodynamics forces and moments such as into:

\[
F_{\text{aero|W, B}} = \frac{1}{2} \rho V^2 S \begin{bmatrix} C_x & C_y & C_z \end{bmatrix}^T
\]

(3.24)

\[
M_{\text{aero|W, B}} = \frac{1}{2} \rho V^2 S \begin{bmatrix} b C_l & \bar{c} C_m & b C_n \end{bmatrix}^T
\]

(3.25)

where the non-dimensional body-axis coefficients are obtained from the AVL analysis with user-specified trimmed flight conditions with \( \alpha = \tan^{-1} \frac{u}{w} \) and \( \beta = \sin^{-1} \frac{v}{V} \). Assuming levelled flight with no sideslip \( \beta \), the lateral body-axis coefficients \( C_y = C_l = C_n = 0 \), while the longitudinal body-axis coefficients \( C_x, C_z \) can be found in Appendix B.

At tilt-angles (\( \gamma \)) less than 45°, the aerodynamics analysis assumes no lift or moment contributions from the vertical tails as AVL does not support analysis at high angles of attack or sideslip angles resulting from the tilt-angle of the quadrotor frame. Otherwise, the vertical tails are defined as a separate AVL input file, where the sideslip angle and angle of attack seen by the vertical tails are described by:

\[
\begin{bmatrix} u \\ v \\ w \end{bmatrix}_Q = \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{bmatrix} C_{\text{AVL}} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u \sin \gamma - w \cos \gamma \\ -u \cos \gamma - w \sin \gamma \\ v \end{bmatrix}
\]

(3.26)

\[
\alpha_Q = \tan^{-1} \frac{v}{u \sin \gamma - w \cos \gamma}
\]

(3.27)

\[
\beta_Q = \sin^{-1} \frac{-u \cos \gamma - w \sin \gamma}{v}
\]

(3.28)

where \( C_{\text{AVL}} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix} \) is the rotation matrix to realign the results from AVL due to the definition of the orientation of the vertical tails in the input file as shown in Appendix A.1. The sequences of rotation to arrive at Eq. (3.26) are shown in Figure 3.4.

Similar to the AVL analysis of the wing-body, the C.G. of the vertical tail is defined such that the forces and moments will be calculated about the origin of \( \mathbf{F}_{\text{Q|c}} \).

\[
F_{\text{aero|VT, Q|c}} = \frac{1}{2} \rho V^2 S_{VT} C_{\text{AVL}} \begin{bmatrix} C_x|VT & C_y|VT & C_z|VT \end{bmatrix}^T
\]

(3.29)

\[
M_{\text{aero|VT, Q|c}} = \frac{1}{2} \rho V^2 S_{VT} C_{\text{AVL}} \begin{bmatrix} b_{VT} C_l|VT & \bar{c}_{VT} C_m|VT & b_{VT} C_n|VT \end{bmatrix}^T
\]

(3.30)

With the assumption of no sideslip (\( v = 0, \beta = 0 \)) seen by the wing-body, the angle of attack seen by the vertical tail \( \alpha_Q \) is zero. Hence, the longitudinal body-axis coefficients \( C_x|VT = \)

...
Figure 3.4: Derivation of the Angle of Attack and Sideslip Angle of the Vertical Tails

\[ C_{z|VT} = C_{m|VT} = 0, \] while the lateral body-axis coefficients \( C_{y|VT}, C_{l|VT}, \) and \( C_{n|VT} \) can be found in Appendix B.

The aerodynamic forces and moments of the quadrotor-body frame are transformed into \( \vec{F}_B \) to be summed with those from the wing-body frame.

\[
F_{\text{aero}|VT, B} = C_{BQ} F_{\text{aero}|VT, Q|c} \tag{3.31}
\]
\[
M_{\text{aero}|VT, B} = (r_{B|c}) \times F_{\text{aero}|VT, B} + C_{BQ} M_{\text{aero}|VT, Q|c} \tag{3.32}
\]

where \( r_{B|c} \) is calculated according to Eq. (3.23).

**Gravity**

The gravitational force is shown in Eq. (3.33). There is no moment associated with gravity as the dynamics of the UAV are expressed at its C.G., as described in Section 3.3.

\[
F_{\text{grav}, B} = C_{BE} \begin{bmatrix} 0 & 0 & m_g \end{bmatrix}^T \tag{3.33}
\]

**Gyroscopic Couples**

The reactionary gyroscopic couples are additional moments due to the angular momentum of spinning elements, such as the propellers relative to the body axis, which result in an additional term of \( \omega_B \times h' \) expressed as a negative contribution term to \( \vec{G}_B \) in Eq. (3.3).

The angular momentum and the torque generated by the propellers, with the moment of inertia \( I_{\text{prop}} \) and rotational speed \( \Omega_{\text{prop}} \), are expressed as:

\[
h'_{Q} = \begin{bmatrix} 0 \\ 0 \\ I_{\text{prop}}(-\Omega_{\text{prop},1} - \Omega_{\text{prop},2} + \Omega_{\text{prop},3} + \Omega_{\text{prop},4}) \end{bmatrix} \tag{3.34}
\]
\[
h'_{B} = C_{BQ} h'_{Q} \tag{3.35}
\]
\[
M_{\text{gyro}, B} = -\omega_B \times h'_B \tag{3.36}
\]
The angular accelerations of the propellers are considered negligible due to the low rotational inertia of the propellers and the relatively constant rotational speed.

The torque required to apply angular acceleration to the quadrotor-body frame is expressed in Eq. (3.37), with the counteracting torque being the negative of the torque applied by the servos. This counteracting torque is added to the \( \mathbf{G}_B \) in Eq. (3.3). Note that the tilt-angle \( \gamma \) increases in the negative direction of the \( y \)-axis, hence the negative sign preceding the term \( \vec{y}_B \).

\[
\mathbf{\tau}_{\text{servo}} = -J_{T,y} \dot{\gamma} \quad \text{(in the direction of } \vec{y}_B) \tag{3.37}
\]

\[
J_T = \mathbf{I}_T + m_T \left[(\mathbf{r}_Q|_{\text{hinge}}) \cdot (\mathbf{r}_Q|_{\text{hinge}}) \mathbf{1}_{3 \times 3} - (\mathbf{r}_Q|_{\text{hinge}})(\mathbf{r}_Q|_{\text{hinge}})^T\right]
\]

\[
\mathbf{r}_Q|_{\text{hinge}} = -\begin{bmatrix} x_T & 0 & z_T \end{bmatrix}^T
\]

\[
\mathbf{M}_{\text{servo}, B} = -\mathbf{\tau}_{\text{servo}} \tag{3.38}
\]

The reactionary gyroscopic coupling of tilting the spinning propellers is expressed in Eq. (3.39). The angular momentum of the spinning propellers, \( \mathbf{h}'_B \), in the wing-body reference frame, is described above in Eq. (3.35). The tilt-angle \( \gamma \) increases in the negative direction of the \( y \)-axis, hence the negative sign preceding the angle.

\[
\mathbf{M}_{\text{tilt}, B} = -\left(\mathbf{\omega}_{\text{tilt}, B}\right) \times \mathbf{h}'_B = -\begin{bmatrix} 0 \\ -\dot{\gamma} \\ 0 \end{bmatrix}^\times \mathbf{h}'_B \tag{3.39}
\]

The moments \( \mathbf{M}_{\text{gyro, } B} \) and \( \mathbf{M}_{\text{tilt, } B} \) can be represented in the quadrotor-body reference frame by multiplying them with the rotation matrix \( \mathbf{C}_{QB} \). The torque to apply angular acceleration to the wing-body frame is expressed in Eq. (3.40).

\[
\mathbf{\tau}_{\text{servo}} = \mathbf{J}_W \dot{y} \quad \text{(in the direction of } \vec{y}_B) \tag{3.40}
\]

\[
\mathbf{J}_W = \mathbf{I}_W + m_W \left[(\mathbf{r}_B|_{\text{hinge}}) \cdot (\mathbf{r}_B|_{\text{hinge}}) \mathbf{1}_{3 \times 3} - (\mathbf{r}_B|_{\text{hinge}})(\mathbf{r}_B|_{\text{hinge}})^T\right]
\]

\[
\mathbf{r}_B|_{\text{hinge}} = -\begin{bmatrix} (h_T - h_W) \bar{c} & 0 & 0 \end{bmatrix}^T
\]

\[
\mathbf{M}_{\text{servo, } Q} = -\mathbf{\tau}_{\text{servo}} \tag{3.41}
\]
Chapter 4

Control of the AmphiQuad UAV

This chapter describes the altitude and attitude controllers proposed for the UAV to be used throughout the transition from hover to forward flight. The design of a gain-scheduled altitude controller using the linearized longitudinal dynamic model of the UAV is presented, with the gains determined through pole placement or the LQR method. A quaternion-based attitude controller based on Reference [58] is described to track the desired pitch angle of the quadrotor-body when transitioning from hover ($\theta_Q = 0^\circ$) to forward flight ($\theta_Q = -90^\circ$) using a continuously differentiable Sigmoid function.

4.1 Gain-Scheduled Altitude Controller

4.1.1 Simplified Longitudinal Dynamics of UAV

The dynamics of the UAV presented in Chapter 3 are highly nonlinear and contain coupling between the rotational dynamics and translational dynamics of the UAV, which make controller design challenging. Several simplifying assumptions were made to develop the linear altitude controller:

- the aerodynamics are represented using the lift ($C_L$) and drag ($C_D$) coefficients;
- the UAV is in level flight and has a zero flight path angle;
- the angle of attack of the wing is maintained by manipulating the orientation of the wing $\theta_B$ through the tilt-servo, hence $C_{ZE} = -C_L$ and $C_{XE} = -C_D$ are constant, and the Reynolds number has no effect on the aerodynamic coefficients;
- the wind speed is assumed to be zero, hence ground speed equals airspeed; and
- the effects of sideslip are considered negligible.

As shown in Figure 4.1, the longitudinal dynamics of the UAV are written in the inertial frame of reference ($x$-axis points forward, $z$-axis points downward) as,

$$\dot{x}_E = \frac{1}{m} \left( -\frac{1}{2} \rho (x_E^2 + z_E^2) SC_D - T \sin \theta_Q \right) \quad (4.1)$$

$$\dot{z}_E = \frac{1}{m} \left( W - \frac{1}{2} \rho (x_E^2 + z_E^2) SC_L - T \cos \theta_Q \right) \quad (4.2)$$

37
Chapter 4. Control of the AmphiQuad UAV

Figure 4.1: Longitudinal Dynamics of the UAV in the Inertial Frame of Reference

The control variables are the collective thrust output of the motors ($T$) and the elevons deflection angle ($\delta_e$). Deflecting the elevons result in a change in the aerodynamic coefficients that are reflected in the $C_{L}$ and $C_{D}$ terms. Note that the pitch angle of the quadrotor-body ($\theta_Q$) used in developing the controllers in the quadrotor-body frame is equivalent to the tilt-angle $\gamma = \theta_B - \theta_Q + \gamma_0$, where $\gamma_0$ is the incidence tilt-angle to achieve the desired angle of attack ($\alpha$).

4.1.2 Equilibrium Conditions

When the UAV is in an unaccelerated level flight at a constant angle of attack, the equilibrium conditions are,

$$0 = \frac{1}{m} \left( -\frac{1}{2} \rho \dot{x}_E^2 SC_D - T \sin \theta_Q \right)$$  \hspace{1cm} (4.3)

$$0 = \frac{1}{m} \left( W - \frac{1}{2} \rho \dot{x}_E^2 SC_L - T \cos \theta_Q \right)$$

Hence, the equilibrium thrust and airspeed are calculated at each $\theta_Q$ to be,

$$v_{x0} = \dot{x}_E_0 = \sqrt{\frac{mg}{\frac{1}{2} \rho S (C_L - C_D \cot \theta_Q)}}$$ \hspace{1cm} (4.4)

$$T_0 = \sqrt{T_{x0}^2 + T_{z0}^2} = \sqrt{(\frac{1}{2} \rho v_{x0}^2 SC_D)^2 + (\frac{1}{2} \rho v_{x0}^2 SC_L - mg)^2}$$ \hspace{1cm} (4.5)

The equilibrium horizontal speed and thrust are shown in Figure 4.2 and 4.3 as a function of the pitch angle of the quadrotor-body (ranging from $\theta_Q = -10^\circ$ to $-90^\circ$) and angle of attack of the wing-body (ranging from $\alpha = 2^\circ$ to $15^\circ$). Note that Figures 4.2 and 4.3 are color contour plots; the color bar on the right of the figure indicates the values of the variables in the title of the figure as a function of the variable in the x-axis and y-axis.
4.1.3 Controller Design

The longitudinal dynamics presented in Eq. (4.1) are linearized at a given desired pitch angle of the quadrotor-body frame, $\theta_Q$, around the equilibrium conditions where $\ddot{x}_E = \ddot{z}_E = \dot{z}_E = 0$.

$$\ddot{v}_x = \Delta \ddot{x}_E = \frac{1}{m} \left(- (D_0 + \Delta D) - (T_0 + \Delta T) \sin \theta_Q \right)$$

$$= \frac{1}{m} \left(- D_0 - T_0 \sin \theta_Q \right) - \Delta D - \Delta T \sin \theta_Q$$

$$D_0 + \Delta D = \frac{1}{2} \rho (V_0 + \Delta V)^2 \sin \left(C_{D_0} + \Delta C_D \right)$$

$$= \frac{1}{2} \rho V_0^2 \sin \left(C_{D_0} + \Delta C_D \right) + \rho V_0 \Delta \cos \theta_Q \Delta C_D$$

$$\Delta D = \frac{1}{2} V_0^2 \sin \left(C_{D_0} + \Delta \cos \theta_Q \right) + \rho V_0 \Delta \cos \theta_Q \Delta C_D$$
Similarly, Eq. (4.2) is linearized to give,

$$
\dot{v}_z = \Delta^2 \varepsilon = \frac{1}{m} (W - (L_0 + \Delta L) - (T_0 + \Delta T) \cos \theta_Q)
$$

$$(4.7)$$

$$
= \frac{1}{m} \left( W - L_0 - T_0 \cos \theta_Q \right) - \Delta L - \Delta T \cos \theta_Q = 0
$$

$$L_0 + \Delta L = \frac{1}{2} \rho (V_0 + \Delta V)^2 S (C_{L,0} + \Delta C_L)
$$

$$= \frac{1}{2} \rho (V_0^2 + 2V_0 \Delta V + \Delta V^2) S (C_{L,0} + C_{L,\delta \varepsilon} \delta \varepsilon)
$$

$$= \frac{1}{2} \rho V_0^2 S C_{L,0} + \frac{1}{2} \rho V_0^2 S C_{L,\delta \varepsilon} \delta \varepsilon + \rho V_0 \Delta V S C_{L,0} + \text{2nd order terms}
$$

$$\Delta L = \frac{1}{2} \rho V_0^2 S C_{L,\delta \varepsilon} \delta \varepsilon + \rho V_0 \Delta V S C_{L,0}
$$

Although the commanded change in thrust from the equilibrium value, \(\Delta T\), is a scalar quantity, it is decomposed into \(x\) and \(z\) components using the linearized equation,

$$
\Delta T \approx \frac{\partial T}{\partial x} \Delta T_x + \frac{\partial T}{\partial z} \Delta T_z
$$

$$
= \frac{1}{2} (T_{x0}^2 + T_{z0}^2) \Delta T_x + \frac{1}{2} (T_{x0}^2 + T_{z0}^2) \Delta T_z
$$

$$= \frac{T_{x0}}{I_0} \Delta T_x + \frac{T_{z0}}{I_0} \Delta T_z
$$

$$(4.8)$$

The linearized equations are then written in state space form,

$$
\begin{align*}
\begin{bmatrix}
\dot{v}_x \\
\dot{v}_z \\
\dot{z}
\end{bmatrix}
&= \frac{1}{m} \begin{bmatrix}
-\rho S_{wing} v_{x0}(\theta_Q) C_D & 0 & 0 \\
-\rho S_{wing} v_{x0}(\theta_Q) C_L & 0 & 0 \\
0 & m & 0
\end{bmatrix} \begin{bmatrix}
\dot{v}_x \\
\dot{v}_z \\
z
\end{bmatrix} + ... \\
&= \frac{1}{m} \begin{bmatrix}
-T_{x0} \sin \theta_Q & -T_{z0} \sin \theta_Q & 0 \\
-T_{x0} \cos \theta_Q & -T_{z0} \cos \theta_Q & 0 \\
0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
\Delta T_x \\
\Delta T_z \\
\delta \varepsilon
\end{bmatrix}
\end{align*}
$$

$$(4.9)$$

where \(T_{x0} \geq 0, \ T_{z0} \leq 0\) and \(\Delta \theta = \Delta V = v_x - v_{x0}\). The drag contribution from the elevon deflection \(C_{D,\delta \varepsilon}\) is dropped due to its nonlinear and negligible effects on the horizontal speed.

Decomposing the scalar thrust according to Eq. (4.8) allows the altitude controller to be used to maintain altitude even when the UAV is in hover mode with \(\theta_Q = 0^\circ\). When the equilibrium horizontal thrust, \(T_{x0}\), is zero, the error in horizontal velocity does not contribute to a change in the overall thrust. This can be seen in Eq. (4.9), where the \(\Delta T_x\) term is nullified by multiplying with a zero value of \(T_{x0}\) to determine the change in thrust required from the equilibrium thrust. On the other hand, during forward flight where \(\theta_Q = -90^\circ\), the \(\Delta T_z\) term in Eq. (4.9), which minimizes the UAV altitude error, is nullified by multiplying a zero value of \(T_{z0}\).

A gain-scheduled state feedback control is proposed with the pitch angle \((\theta_Q)\) being the scheduling variable, as shown in Eq. (4.10). The system is not fully controllable when the UAV is hovering with pitch angle, \(\theta_Q = 0^\circ\), as the propellers are at zero pitch angle and elevons
at zero airspeed are incapable of controlling the horizontal speed of the UAV. The altitude controller hence reduces into a PD controller with the output thrust based on the climb speed and error in altitude. For pitch angles of the quadrotor-body in the range of $-90^\circ \leq \theta_Q < 0^\circ$, the system is controllable with full rank of 3.

$$\begin{bmatrix} \Delta T_x \\ \Delta T_z \\ \delta_e \end{bmatrix} = \begin{bmatrix} -k_{v_x}(\theta_Q) & 0 & 0 \\ 0 & -k_{v_z}(\theta_Q) & -k_z(\theta_Q) \\ k_{\delta_e,v_z}(\theta_Q) & k_{\delta_e,z}(\theta_Q) \end{bmatrix} \begin{bmatrix} \dot{\theta}_x \\ v_z \\ z \end{bmatrix}$$ (4.10)

where the feedback gains are: $k_{v_x}, k_{v_z}, k_z, k_{\delta_e,v_z}, k_{\delta_e,z} \in \mathbb{R} > 0$.

In designing the altitude controller, a simplified MATLAB Simulink model was constructed to simulate the longitudinal dynamics in Eqs. (4.1) and (4.2), while assuming that the pitch angle of the quadrotor frame ($\theta_Q$) perfectly tracks Eq. (4.11) for transitioning from hover to forward flight.

$$\theta_{Q,d}(t) = (\theta_{Q,f} - \theta_{Q,0}) \frac{1}{1+\exp \left( \frac{-a}{\theta_{Q,0} - t - c} \right)} + \theta_{Q,0} \text{ rad}$$ (4.11)

where $a$ represents the rate of change in the desired pitch angle and $c$ is the time to the transition mid-point.

Figure 4.4: Block Diagram of the Simulink Model for Longitudinal Dynamics of the UAV

As shown in Figure 4.4, the altitude controller takes in the pitch angle, climb rate, altitude, and horizontal speed of the quadrotor-body, and runs the altitude controller in Eq. (4.10). The outputs of the altitude controller block are the desired thrust force of the motors and elevator deflection angle; these are then fed into the longitudinal dynamic model to obtain the linear accelerations which are then numerically integrated to obtain the translational velocities and altitude of the UAV which form the feedback signals into the controller. The simulation was conducted using the ode45 (Dormand-Prince) solver with an automatic variable-time step.

The simulation was run with the pitch angle of the quadrotor-body changing from $-5^\circ$ to $-90^\circ$ following the Sigmoid function in Eq. (4.11), with $a=0.25$ and $c=50$. The initial condition was set to have the UAV at a horizontal speed of 5 m/s, which simulates the airspeed gained by the UAV when quadrotor frame is pitched downwards before the transition begins. Note that the equilibrium speed at $-5^\circ$ pitch angle of the quadrotor-body is not necessarily 5 m/s,
and is dependent on the angle of attack (hence the lift coefficient) of the UAV.

### 4.1.4 Determining Gains of Altitude Controller

#### Pole Placement

To ensure the stability of the closed-loop system shown in Eq. (4.9) at all pitch angles of the quadrotor-frame $θ_Q$, the gain matrix $F$ was determined such that the close-loop poles are placed in the left-half of the s-plane. To ensure such a condition is met, the MATLAB `place` function \[59\] was utilized to determine the gain matrix as a function of $θ_Q$. To maximize range of the UAV, we chose to fly around 14 m/s which corresponds to an angle of attack of $8^\circ$. The lift and drag coefficients at this angle of attack ($C_L = 0.459$, $C_D = 0.043$, $C_{L,δ_e} = 0.6825/rad$) were used to construct the $A$ matrices in Eq. (4.9).

![Figure 4.5: Elements of Gain Matrix $F$ at Each Quadrotor Pitch-Angle $θ_Q$ Using Pole Placement](image)

Different pole locations were chosen with the elements of the gain matrix being shown in Figure 4.5, where increasingly negative pole placement results in higher gain values. For a chosen pole location, the gains $k_{vx}$, $k_{δ_e,vz}$, and $k_{δ_e,z}$ are relatively constant across all pitch angles of the quadrotor-body frame, suggesting that gain-scheduling may only be necessary for the $ΔT_z$ response on the climb rate and error in altitude. On the other hand, the $k_{vz}$ and $k_z$ gains increase significantly towards zero pitch angle of the quadrotor-body (hovering state), where low airspeed near hover reduces the effectiveness of the elevons and hence the controller relies more heavily on the vertical thrust to maintain the altitude of the UAV.

The simulation results show that a more negative pole location results in a larger elevon deflection during the transition which reduces the change in altitude. However, the higher gains as a result of a more negative pole location cause oscillations in the control outputs. Selecting the pole location at -2 appears to be a good compromise between performance and control efforts.
Figure 4.6: Simulink Results for Longitudinal Dynamics with Gain Scheduled Altitude Controller Using Pole Placement Method With Different Pole Locations

Linear Quadratic Regulator

In the Linear Quadratic Regulator (LQR) method, the control gains were selected to minimize a cost function ($J$) which is a function of the error in the states and the control outputs [60],

$$J = \frac{1}{2}x^T(t_f)x(t_f) + \int_0^{t_f} \left[ \frac{1}{2}(x^TQx + u^TRu) \right] dt$$ (4.12)

In contrast to pole placement, the LQR method allows fine-tuning of the balance between the controller performance and the control output to prevent saturation, as well as providing the flexibility to specify the weights of the states and control outputs. When choosing the pole locations using pole placement, it is unclear how each pole correlates which the convergence performance of each state variable. The LQR method, on the other hand, allows assignment of weights on the state variable through the matrix $Q$.

The tuneable parameters are the symmetric positive definite matrices $S$, $Q$, and $R$, where a higher $Q/R$ ratio results in a faster convergence at the expense of larger control outputs. Obtaining the gain matrix $F$ involves solving the algebraic Riccati equation [60, 61],

$$PA + A^T\dot{P} - \dot{P}BR^{-1}B^T\dot{P} + Q = 0$$ (4.13)

The MATLAB care function [61] was used to obtain the $\bar{P}$ matrix and hence the gain matrix,

$$F = -R^{-1}B^T\bar{P}$$ (4.14)
Figure 4.7: Simulink Results for Longitudinal Dynamics with Gain Scheduled Altitude Controller Using LQR With Different \( Q \) And \( R \) Matrices

The simulation results in Figure 4.7 show the effects of varying the matrix \( Q \) and \( R \) on the errors in the state variables and the control outputs, based on the following four cases:

1. \( Q = 0.25 \cdot I_{3 \times 3}, R = 1 \cdot I_{3 \times 3} \)
2. \( Q = 0.75 \cdot I_{3 \times 3}, R = 1 \cdot I_{3 \times 3} \)
3. \( Q = 0.25 \cdot \text{diag}(1,1,5), R = 1 \cdot I_{3 \times 3} \)
4. \( Q = 0.25 \cdot I_{3 \times 3}, R = 1 \cdot \text{diag}(1,1,5) \)

Increasing \( Q \) by an overall factor (case 2) results in control saturation, as observed in the elevon deflection angle at the beginning of the transition. This has a similar effect when increasing the diagonal terms in the \( Q \) matrix (case 3). As a higher priority is given to the altitude error instead of the velocity errors, a more aggressive elevon response is observed near hover condition, but it does not help to improve altitude regulation due to its low effectiveness at low airspeed. Changing \( R \) to penalize large elevon output (case 4) helps to reduce its deflection angle at the expense of higher thrust output.

While the altitude change throughout the transition is larger with the gains determined using the LQR method as compared to that using pole placement, the LQR method provides an easier method to understand and tune the gains.

**Discarding Terms in the Gain Matrix**

The gain matrix \( F \) shown in Eq. (4.10) contains zero terms as the thrust required in the \( x \)-direction should not depend on the velocity in \( z \)-direction and vice versa. This is equivalent to a P-control for the forward speed and a PD-control for the altitude. However, when using pole placement and the LQR method, the gain matrix \( F \) does not have those zero terms, due to the coupling between horizontal velocity and climb rate (hence altitude) through the lift
generated by the wing. Replacing the elements of the gain matrix $F$ determined through pole placement and the LQR method with zeros as shown in Eq. (4.15) are necessary to prevent the control outputs from being too sensitive to the coupling errors, but doing so will affect the eigenvalues and might change the stability of the closed-loop system.

$$
\begin{bmatrix}
  k_{11} & k_{12} & k_{13} \\
  k_{21} & k_{22} & k_{23} \\
  k_{31} & k_{32} & k_{33}
\end{bmatrix}
\rightarrow
\begin{bmatrix}
  -k_{v_1} & 0 & 0 \\
  0 & -k_{v_2} & -k_z \\
  0 & k_{\delta v_i v_z} & k_{\delta v_i z}
\end{bmatrix}
$$

(4.15)

Figure 4.8: Eigenvalues of the Closed-Loop System with Gains Using Pole Placement Method

Figure 4.9: Eigenvalues of the Closed-Loop System with Gains Using LQR Method

Figures 4.8 and 4.9 show the eigenvalues of the $A + BF$ matrix with and without the zero terms in the $F$ matrix determined using pole placement (with poles at -3) and the LQR method (with $Q = 10 \cdot \text{diag}(1,1,10)$, $R = 0.1 \cdot \text{diag}(1,1,100)$ ) respectively. The left figure shows the eigenvalues of the closed-loop system with the full gain matrix, while the right figure shows the eigenvalues of the closed-loop system with the zeros in the gain matrix. As observed in Figure 4.8, the deviation from the desired pole location of -3 towards zero occurs as $\theta_Q$ is approaching $0^\circ$ (hover), but the closed-loop system remains stable. On the other hand, the eigenvalues of the closed-loop system with the gains determined through the LQR method as shown in Figure 4.9 see very little change with and without the zero terms, due to the mostly negligible $k_{12}, k_{13}, k_{21},$ and $k_{31}$ terms in the $F$ matrix.
4.1.5 Equilibrium Speed Update

The equilibrium speed was calculated in Section 4.1.2 with the assumption that the aerodynamic coefficients are constant and are known. In practice, these coefficients are subjected to modelling errors and can vary in different operating conditions, such as a change in the flow Reynolds number, the presence of wind, and the variation in air density due to ambient temperature. The UAV has no means of measuring and maintaining those desired aerodynamic coefficients. In addition, controlling the altitude of the UAV through elevon deflections causes further deviation in the actual lift and drag coefficients from those values used for calculating the equilibrium speed.

To address such deviation in the equilibrium speed, the equilibrium speed is updated based on the vertical acceleration and climb rate of the UAV,

\[ \dot{v}_{x0} = \sigma_1 \dot{v}_z + \sigma_2 v_z \]  

where \( \sigma_1, \sigma_2 > 0 \). If the actual lift coefficient is lower than the predicted equilibrium value, the equation will increase the equilibrium speed when the UAV loses altitude, and vice versa.

![Simulink Results](image)

**Figure 4.10:** Simulink Results for Longitudinal Dynamics with Gain Scheduled Altitude Controller Using Pole Placement Method With and Without the Speed Update Equation

To observe the advantages of such update equation, the simulation was run with a different lift and drag coefficients than those used for calculating the equilibrium conditions. The results for a transition trajectory from hover (0° pitch) to forward flight (-90° pitch) are shown in Figure 4.10 where the actual lift coefficient \( C_L \) is reduced by 15% and the actual drag coefficient \( C_D \) is increased by 20%. With the speed update equation, the cruising speed of the UAV increases from 14.5 m/s to 15.3 m/s to account for the decrease in lift coefficient, and the elevon deflection occurs earlier during the transition to compensate for the descent between 20
s to 35 s of the simulation. As a result, the altitude change at the end of the transition with the speed update equation is less than that without the equation, in addition to the lower control effort required by elevons.

### 4.1.6 Effects of Variation in Transition Trajectory

While the gains of the altitude controller determined through pole placement and the LQR method guarantee stability of the linearized closed-loop system at each pitch-angle increment of the quadrotor-body frame, the stability of the system with varying pitch-angle throughout the transition is not guaranteed. The scheduling variable (the pitch angle of the quadrotor frame) “should vary slowly” to ensure stability of the gain-scheduled control [62].

While Reference [63] outlined the method to determine an upper bound for the variation of the scheduling variable to ensure exponential stability of the linearized system, the method only applies to a specific state-space interpolation method presented in the paper. Due to time constraints and the focus on the implementation and experimentation parts of the research, this interpolation method was not explored. Instead, a simple linear interpolation is used to determine the gain values in between the pitch increments.

---

**Figure 4.11:** Real Part of the Eigenvalues of the $A + BF$ Matrix with Pole Placement Gains

**Figure 4.12:** Real Part of the Eigenvalues of the $A + BF$ Matrix with LQR Gains
The stability of the system with the gain matrix determined using pole placement and the LQR method is shown in Figure 4.11 and 4.12 respectively, where the stability of the system was analyzed through the eigenvalues of the matrix $A(\theta_Q) + B(\theta_Q) F(\theta_{ref, Q})$. The matrix $F(\theta_{ref, Q})$ was determined for a specific pitch angle $(\theta_{ref, Q})$, while the $A(\theta_Q)$ and $B(\theta_Q)$ matrices were evaluated at different pitch angles of the quadrotor-body $(\theta_Q)$. The analysis shows that while the eigenvalues change considerably, the linearized system remains stable when the gains determined for a specific pitch angle of the quadrotor-body $(\theta_{ref, Q})$ are used at a different pitch angle $(\theta_Q)$.

The effects of varying the transition trajectory were studied in the simulation results shown in Figure 4.13. The desired pole locations of the altitude controller were chosen to be -2 throughout the simulations. The Sigmoid function shown in Eq. (4.11) was initially chosen due to its continuous and gradual change in the pitch angle as a function of time, which allows it to be differentiable and used for feedforward of the angular velocity in the attitude tracking controller described in Section 4.2. A larger $a$ in the Sigmoid function equates to a faster transition, which then results in a more aggressive response in the control outputs as the actuators have to respond more quickly to the increase in desired horizontal speed. In such a situation, the UAV does not have sufficient speed to generate lift and the thrust vector is tilted downwards too quickly to generate thrust in the vertical direction, which leads to slower response in the altitude regulation.

The simulation was repeated with varying slopes of the ramp transition trajectory from -5° to -90° pitch angle of the quadrotor-body, with the start of the transition at $t=25$ s. As shown in Figure 4.14, the different slopes do not appear to affect the performance of the altitude controller to regulate altitude. This can be attributed to the perfect pitch tracking assumption in this simplified simulation model, as a delay in the pitch response of the UAV could lead to
more drastic altitude variations.

Figure 4.14: Simulink Results for Longitudinal Dynamics with Gain Scheduled Altitude Controller Using PP Method With Different Transition Trajectory (Ramp Functions)

A step function was also considered for the transition trajectory, and was implemented in the simulation through the Simulink Step Function block. Two cases were considered: the first involves stepping from -5° to -90° pitch angle at 25 s, and the second involves stepping from -5° to -45° at 25 s and then to -90° at 50 s.

Figure 4.15: Simulink Results for Longitudinal Dynamics with Gain Scheduled Altitude Controller Using PP Method With Different Transition Trajectory (Step Functions)

For the first case, as shown in Figure 4.15, the thrust output at the end of the transition is at 53.4 N with a horizontal speed of 32 m/s with the elevon deflection upwards at 30.4°. The equilibrium achieved is not efficient, and is practically not achievable due to insufficient
dynamic thrust at that airspeed. It is a result of the speed update equation (Section 4.1.5) calculating a much higher equilibrium speed as the UAV experiences a drop in altitude, which is caused by insufficient airspeed to generate lift when the pitch angle changes too abruptly. The two-step pitch angle approach in the second case shows a successful transition to forward flight but with some noticeable oscillations at those step times.

The simulation results suggest that the ramp or Sigmoid function should be preferred over step functions, as the sudden jump introduced by the step function in the desired velocity could lead to undesirable effects on the control outputs and hence the equilibrium achieved by the system. While the simulation does not lead us to define a upper limit on the variation rate of the pitch angle, a Sigmoid function with $a < 0.5$, or a ramp function with slope $< 5^\circ/s$ generally provides good controller performance with reasonable control efforts.

### 4.2 Quaternion-based Attitude Controller

When using the quadrotor-body as the frame of reference to develop the attitude controller for the UAV, the pitch angle of the quadrotor-body will reach $-90^\circ$ at the end of the transition to forward flight. This will result in singularity in the attitude representation as shown in Eq. (3.10) if the autopilot board is mounted on the quadrotor-body and. To ensure a singularity-free controller, quaternions were chosen to represent the UAV attitude and to develop the attitude tracking controller. The UAV is capable of producing torques in all three body axes by differing the thrust outputs on each of the four motors, hence allowing control of all three rotational degrees of motion. During transition, the pitch angle of the quadrotor-body is commanded to follow a trajectory from $0^\circ$ to $-90^\circ$, while the roll and yaw angles are kept at zero. The desired attitude is represented in quaternions $(\epsilon_d, \eta_d)$ with the rotation angle $\Phi(t) : 0^\circ \rightarrow -90^\circ$ along the rotation axis $a = [0; 1; 0]$.

As quaternion-based attitude control is not common in quadrotors due to the assumption that they operate near hover condition, the method from Reference [58] was used to implement a quaternion-based attitude tracking controller on the AmphiQuad UAV. The controller is globally asymptotically stable, free from singularities, and does not require any prior knowledge of the mass or inertia of the vehicle [58]. The adaptive inertia matrix update law presented in Reference [58] was not used, as the UAV is not expected to undergo aggressive maneuvers that require accurate knowledge of the inertia matrix. The rotation matrix is related to the quaternions through the equation,

\[ C(\epsilon, \eta) = (1 - 2\epsilon^T \epsilon)1 + 2\epsilon \epsilon^T - 2\eta \epsilon^\times \]

\[ C_{QD} = C(\epsilon_d, \eta_d) = C(\epsilon, \eta)C(-\epsilon_d, \eta_d) \]
The quaternion error \((\mathbf{e}_e, \eta_e)\) between the desired attitude and the current attitude of the UAV is represented in the equation,

\[
\mathbf{e}_e = \eta_d \mathbf{e} - \eta \mathbf{e}_d - \mathbf{e}_d^T \mathbf{e} \quad (4.18)
\]

\[
\eta_e = \mathbf{e}_d^T \mathbf{e} + \eta \eta_d
\]

The error in angular velocity of the quadrotor frame is represented as a relative angular velocity of the quadrotor frame \(\mathbf{F}_Q\) to the desired frame \(\mathbf{F}_D\) as shown in Eq. (4.19).

\[
\mathbf{\omega}_Q^D = \mathbf{\omega}_Q^E - \mathbf{\omega}_D^E \quad (4.19)
\]

Assuming all the moments presented in Chapter 3 (except the moments from the propulsion system) are lumped into a disturbance term \(d\), the rotational dynamics become,

\[
\dot{\mathbf{J}} \mathbf{\omega}_Q^D = \mathbf{J}((\mathbf{\omega}_Q^D)^T \mathbf{C}_{QD} \mathbf{\omega}_D^E - \mathbf{C}_{QD} \dot{\mathbf{\omega}}_D^E) - \ldots \quad (4.20)
\]

\[
\ldots (\mathbf{\omega}_Q^D + \mathbf{C}_{QD} \mathbf{\omega}_D^E)^T \mathbf{J}(\mathbf{\omega}_Q^D + \mathbf{C}_{QD} \mathbf{\omega}_D^E) + \mathbf{\tau} + \mathbf{d}
\]

\[
\dot{\mathbf{e}}_e = \frac{1}{2} (\mathbf{e}_d^T \mathbf{\omega}_Q^E + \eta \mathbf{\omega}_Q^D)
\]

\[
\dot{\eta}_e = -\frac{1}{2} \mathbf{\epsilon}_d^T \mathbf{\omega}_Q^D
\]

The inertial matrix \(\mathbf{J} \in \mathbb{R}^{3 \times 3}\) is a symmetric positive definite matrix that can be rewritten as a vector \(\mathbf{\Gamma} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \in \mathbb{R}^{6 \times 1}\). Following Reference [58], the operator \(\mathcal{L} : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 6}\) is defined to handle matrix multiplication between the inertia matrix and a vector, \(\mathbf{v} \in \mathbb{R}^{3 \times 1}\), such that,

\[
\mathbf{Jv} = \mathcal{L}(\mathbf{v}) \mathbf{\Gamma}
\]

where \(\mathcal{L}(\mathbf{v}) = \begin{bmatrix} v_1 & 0 & 0 & v_3 & v_2 \\ 0 & v_2 & 0 & v_3 & v_1 \\ 0 & 0 & v_3 & v_2 & v_1 & 0 \end{bmatrix}\).

Eq. (4.20) is hence rewritten with the \(\mathcal{L}\) operator as,

\[
\dot{\mathbf{J}} \dot{\mathbf{\omega}}_Q^D = \mathbf{G}(\mathbf{\omega}_Q^D, \mathbf{C}_{QD}, \mathbf{\omega}_D^E, \dot{\mathbf{\omega}}_D^E) \mathbf{\Gamma} + \mathbf{\tau} + \mathbf{d} \quad (4.22)
\]

\[
\mathbf{\Gamma} = \begin{bmatrix} \mathcal{L}(\mathbf{K}_1 \mathbf{e}_e) + \mathbf{G}(\mathbf{\omega}_Q^D, \mathbf{C}_{QD}, \mathbf{\omega}_D^E, \dot{\mathbf{\omega}}_D^E) \mathbf{\Gamma} + \mathbf{\tau} + \mathbf{d} \end{bmatrix} \quad (4.23)
\]

\[
\mathbf{G}(\mathbf{\omega}_Q^D, \mathbf{C}_{QD}, \mathbf{\omega}_D^E, \dot{\mathbf{\omega}}_D^E) = \begin{bmatrix} \mathcal{L}((\mathbf{\omega}_Q^D)^T \mathbf{C}_{QD} \mathbf{\omega}_D^E - \mathbf{C}_{QD} \dot{\mathbf{\omega}}_D^E) - \ldots \quad (4.24)
\]

\[
\ldots (\mathbf{\omega}_Q^D + \mathbf{C}_{QD} \mathbf{\omega}_D^E)^T \mathcal{L}(\mathbf{\omega}_Q^D + \mathbf{C}_{QD} \mathbf{\omega}_D^E) \]

where the error is defined as \(\mathbf{\dot{\omega}} = \mathbf{\omega}_Q^D + K_1 \mathbf{e}_e\).

The attitude tracking controller outputs the commanded torque vector (\(\mathbf{\tau}\)) according to,

\[
\mathbf{\tau} = -\mathbf{\mathcal{G}(\mathbf{\omega}_Q^D, \mathbf{C}_{QD}, \mathbf{\omega}_D^E, \dot{\mathbf{\omega}}_D^E) + \mathcal{L}(\mathbf{K}_1 \mathbf{e}_e)} \mathbf{\Gamma} - k \mathbf{e}_e - K_2 \mathbf{\dot{\omega}} - k_3 \int_0^t \mathbf{\dot{\omega}} \, dt \quad (4.24)
\]
where the first term in square bracket represents the feedforward term for desired angular velocities and accelerations, and the remaining terms represent the feedback terms based on the error in quaternions and the relative angular velocity. An additional integral term is included to compensate for $d$. The gains are: $K_1, K_2 \in \mathbb{R}^{3 \times 3}$, $k, k_3 \in \mathbb{R}$ with $K_1 = K_1^T > 0$, $K_2 = K_2^T > 0$, $k, k_3 > 0$. They are determined through trial-and-error by evaluating the performance of the attitude controller in simulation.

**Wing-Body Mounted Autopilot Board**

The attitude controller presented in the earlier section assumed that the autopilot board is mounted on the quadrotor-body frame, where during transition, the desired pitch angle of the quadrotor-body changes from $0^\circ$ (hover) to $-90^\circ$ (forward flight). If we were to fly the UAV using the ArduPlane firmware, the autopilot board has to be mounted on the wing-body frame. In such a situation, the desired pitch angle is $0^\circ$ during hover and is dependent on the flight path angle provided by the ArduPlane altitude controller during forward flight [57].

In such a case, the attitude controller does not require feedforward of the desired angular velocity and acceleration, and the controller presented in Eq. (4.24) is simplified to become,

$$
\tau = -k \varepsilon - K_2 \omega - k_3 \int_0^t \varepsilon dt \tag{4.25}
$$

where $k, k_3 > 0$ and $K_2 = K_2^T > 0$.

In Quadrotor-related (VTOL) flight modes, controlling the position of the UAV involves rolling and pitching the aircraft. As the tilt angle between the quadrotor-body frame and the wing-body frame is fixed (at $\gamma = 0^\circ$, i.e. the chord-line of the vertical tails being perpendicular to the chord-line of the wing-body), pitching the UAV to fly forward creates a nose-down attitude for the wing-body, which creates negative lift and more drag, and subsequently requires the motors to produce more thrust. A solution was proposed to modify the firmware to keep the wing level while changing the tilt-servo angles to provide vectored thrusts for the UAV to fly forward or backward in quadrotor mode. The desired roll, pitch, and yaw moments of the wing-body frame provided by the attitude controller are transformed into the quadrotor-body frame for the tilted motors to provide the correct moments,

$$
\begin{bmatrix}
\tau_x \\
\tau_y \\
\tau_z
\end{bmatrix}_Q =
\begin{bmatrix}
\cos \gamma & 0 & \sin \gamma \\
0 & 1 & 0 \\
-\sin \gamma & 0 & \cos \gamma
\end{bmatrix}
\begin{bmatrix}
\tau_x \\
\tau_y \\
\tau_z
\end{bmatrix}_B \tag{4.26}
$$

During transition, instead of commanding a pitch angle change from $0^\circ$ (hover) to $-90^\circ$ (forward flight), the wing is kept level while the quadrotor-body is rotated from $\gamma = 0^\circ$ to $\gamma = 90^\circ$. Due to the lack of sensors such as encoder and accelerometer in the quadrotor-body frame, the change in tilt-servo angle is an open-loop system.
In Plane-related flight modes, since the chord-line of the wing-body is parallel to the thrust vector and the autopilot is mounted in the wing-body frame, controlling the UAV is similar to that of flying a fixed-wing aircraft with control surfaces. This is unlike the case of mounting the autopilot board on the quadrotor-body frame, where the commanded roll and yaw angles in the quadrotor frame have to be transformed into the wing-body frame before actuating the ailerons.

**Summary**

This section proposed a gain-scheduled altitude controller that was designed based on the simplified and linearized longitudinal dynamic model of the UAV. The gains were determined using pole placement and the LQR method, which eliminated the need for using trial-and-error method to determine the gains at each increment of the scheduling variable. In addition to the thrust output, the controller uses the elevons to improve the altitude regulation throughout transition from hover to forward flight. Additionally, an equilibrium speed update equation was proposed to account for deviations in the equilibrium speed due to modelling errors. The effects of varying the controller gains and transition trajectories were studied with MATLAB Simulink in the simplified simulation model with only the longitudinal dynamics.

This section also proposed the use of a quaternion-based attitude tracking controller that is singularity-free and model-independent. This attitude controller is used when the sensors and autopilot board are mounted on the quadrotor-body frame, and they will tilt forward from a pitch angle of 0° to -90° during the transition from hover to forward flight. The quaternion-based attitude controller, together with the gain-scheduled altitude controller, will be implemented in the MATLAB Simulink model in Chapter 6 to simulate the transition.
Chapter 5

Firmware Development

This chapter describes the work performed on the firmware of the open-source Pixhawk autopilot board which is used to collect and process sensor data, as well as to provide control outputs to fly and stabilize the UAV. The structure of the firmware is explained with comparisons between the ArduCopter and ArduPlane variants of the firmware. This chapter also discusses the QuadPlane feature that is built into the ArduPlane firmware, the implemented transition strategy, and the modifications we proposed to modify the transition strategy. The modifications performed on both variants of the firmware are also documented in this chapter.

5.1 ArduPilot Repository and Structure

As described in Section 2.8, the Pixhawk autopilot board was chosen for its cost effectiveness, wide compatibility with sensors, and continuous support from the open-source community which makes it a suitable platform to modify and develop custom control algorithms.

The Pixhawk autopilot board supports two firmware variants: PX4 and ArduPilot. While the organization of the code is vastly different between the two variants, the functionalities and features they provide for fixed-wing aircraft and multirotors are identical. However, ArduPilot has an advantage over PX4 for supporting more hardware platforms, having more sophisticated flight control algorithms, and demonstrated more reliable results in flight tests [64], and was hence chosen for the development portion of this thesis.

The structure of the ArduPilot repository is shown in Figure 5.1. It consists of libraries that contain codes and algorithms that are vehicle platform-independent, as well as vehicle-specific codes that implement the algorithms available in the libraries on an as needed basis. It also contains external dependencies on other code repositories, such as MavLink for communication protocol, PX4Firmware which contains drivers and middleware for Pixhawk-variant boards, and etc. During compilation, the CMake build system gathers the required libraries and external dependencies to compile the firmware before uploading it to the autopilot board through a USB cable.
Under the GNU Public Licensing of the ArduPilot firmware, all modifications to the original source code are to be made available publicly. The original code repository is available on https://github.com/ArduPilot/ardupilot/; and the changes made for this tilt-rotor platform in this thesis are published at https://bitbucket.org/yihtangyeo/ardupilot_mod/.

ArduCopter

The ArduCopter firmware of the ArduPilot repository supports multirotor-type vehicles such as tricopters, quadrotors, and octorotors. As of version Copter 3.6, it supports multiple flight operation modes such as pilot-controlled stabilized flight, loitering, autonomous waypoint navigation, and autonomous take-off and landing. It also contains optional features such as camera gimbal support, Laser-based altitude hold, and position hold based on optical flow. Using ArduCopter firmware on AmphiQuad UAV involves mounting the autopilot board on the quadrotor frame which also attaches to the motors, and commanding pitch angle downwards to initiate the transition from hover to forward flight.

ArduPlane

ArduPilot, as of version Plane 3.8, supports various types of fixed-wing aircraft ranging from conventional configurations, flying-wings, and canard configurations. In addition, it supports hybrid UAV configurations such as tail-sitters and tilt-rotors through the QuadPlane feature. Using the QuadPlane feature in ArduPlane on AmphiQuad UAV requires that the autopilot board be mounted in the wing-body frame instead of the quadrotor frame.

Default Controllers in ArduPilot Firmware

The existing controllers implemented by the open-source community on the ArduPilot firmware are summarized in Table 5.1. See References [57] and [65] for more information on the algorithms, equations, and implementation details on the controllers.
### Table 5.1: Default Controllers Implemented on ArduCopter And ArduPlane

<table>
<thead>
<tr>
<th>Controller</th>
<th>ArduPlane 3.8</th>
<th>ArduCopter 3.6</th>
</tr>
</thead>
</table>
| Altitude (z-Position) Controller | Total Energy Control System (TECS) based on the kinetic energy and gravitational potential energy of the aircraft.  
  - **PI control**: rate of change in energy difference to elevator output  
  - **PI control**: rate of change in total energy to throttle output | **P control**: position error to target velocity  
  - **P control**: velocity error to acceleration  
  - **PID control**: acceleration error to throttle output |
| Attitude Controller | Use small angle approximation such that angular velocity is equal to the rate of change of Euler angles. Use speed scaler to account for ineffectiveness of control surfaces at low airspeed.  
  - **P control**: attitude error to target angular velocity  
  - **PID control**: angular velocity error to aileron/elevator deflection | Target attitude is transformed into quaternion, and the quaternion error is transformed into thrust vector error (roll/pitch) and heading error (yaw).  
  - **P control**: attitude error to target angular velocity  
  - **PID control**: angular velocity error to motor torques |

### 5.2 Choice of Firmware and Autopilot Mounting Location

### Table 5.2: Comparison Between Modifications On ArduCopter And ArduPlane

<table>
<thead>
<tr>
<th>Location</th>
<th>ArduPlane 3.8</th>
<th>ArduCopter 3.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Wing-Body</td>
<td>Quadrotor-Body</td>
</tr>
<tr>
<td>Transition</td>
<td>Tilt-servo angle changes from 0° to 90°, while the wing-body is kept level by the attitude controller</td>
<td>Quadrotor pitches forward from 0° to -90° while the wing-body is gimballed to stay level</td>
</tr>
<tr>
<td>Benefits</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
  - Built-in QuadPlane feature on ArduPlane serves as a good performance baseline, and require less code modifications  
  - Having the Plane-related flight modes readily-available allow less-risk and ease of testing |  
  - Modifications to ArduCopter code allow the modified firmware to be used on any existing multi-rotor only having to add a lifting surface  
  - Greater familiarity and experience with the features and structures of the ArduCopter firmware |
| Drawbacks |  
  - Less robust state estimation and flight controls than ArduCopter in Quadrotor-related flight modes, especially in the absence of GPS [57]  
  - Lack of safety features that are found in ArduCopter, such as emergency motor stop and assigning return to launch/landing to extra radio channel |  
  - Having to migrate Plane-related flight modes to achieve autonomous forward flying capabilities  
  - Having to migrate Plane-related features such as airspeed sensor reading and L1 waypoint navigation [66]  
  - Having to ensure the state estimation and flight controls do not suffer from singularities in forward flying at $\theta=-90^\circ$ |
Two approaches for making modifications to the ArduPilot firmware were explored: making changes based on the ArduPlane QuadPlane firmware, and adding new flight modes and controllers to the ArduCopter firmware. A comparison of the two approaches, along with their benefits and drawbacks, is shown in Table 5.2.

Initially, the ArduCopter firmware was chosen for implementing the proposed controllers and transition strategy. The motivation for such a decision is that it allows the firmware to be used readily on other multirotor vehicles that already have the autopilot mounted on the frame, which will allow them to enjoy the benefits of having extended range and increased payload capacity through the addition of a lifting surface.

However, due to time and resources constraints, the development work on the ArduCopter firmware was halted due to technical challenges, such as the discrepancies in the propulsion performance in simulation as compared to outdoor test flights, the singularity in the attitude measurement when the pitch angle of the quadrotor frame approaches -90°, and the need to reformulate the waypoint navigation control. These issues will be discussed in Section 7.1.3. As a result, the development work on the firmware was then changed to be based on ArduPlane instead.

5.3 Transition Between Hover and Forward Flight

This section describes the transition strategy implemented by the QuadPlane feature in the ArduPlane firmware, the limitations of such strategy, and the proposed changes to adapt the firmware for the transition of the AmphiQuad UAV.

5.3.1 ArduPlane QuadPlane Transition Strategy

The transition between hovering (Quadrotor mode) and forward-flying (Plane mode) is triggered when the flight mode of the UAV is switched from a Quadrotor-related mode to a Plane-related mode. Some examples of Quadrotor-related modes are QStabilize (ArduCopter Stabilize), QHover (ArduCopter Altitude Hold), QLoiter (ArduCopter Position Hold), and QLand (ArduCopter Vertical Landing), while some examples of Plane-related modes are Manual, Stabilize, Fly-By-Wire A, Fly-By-Wire B, and Auto. Detailed features and differences between the Plane-related flight modes are documented on the developer’s website [57].

For a tilt-rotor like the AmphiQuad UAV, during the transition from quadrotor mode to plane mode, the servos responsible for tilting the motors are rotated by a user-defined amount (default is 45°) at a user-defined rate, and the autopilot then waits for the aircraft to achieve the desired airspeed. This transition state is known as TRANSITION_AIRSPEED_WAIT, which typically also involves an increase in throttle to quickly gain airspeed and to prevent significant altitude loss. Once the desired airspeed is achieved, the transition state enters TRANSITION_TIMER. The motors are fully tilted, the throttle is reduced while the autopilot continues to stabilize the aircraft over a user-defined period. After a predefined amount of time,
the transition is deemed to be completed and the UAV enters the TRANSITION_DONE state.

---

**Figure 5.2: ArduPlane’s QuadPlane Transition Strategy and States Flowchart**

The reverse transition from forward flight to hover involves operating the servo to tilt the quadrotor frame downwards, and allowing the air resistance to automatically slow the UAV down. Unlike the transition from hover to forward flight, there is no in-between states and the tilt-rotor is tilted backward at the user-defined rate in a single action [65]. The altitude and attitude controllers of the Quadrotor-related flight modes are used once the reverse transition begins. Typically, the throttle is reduced to decrease the forward thrust while the attitude is still being maintained by the fixed-wing control surfaces. The transition state is set to TRANSITION_AIRSPEED_WAIT to prepare for future transition into forward flight. The transition and flowchart between the different states are summarized in Figure 5.2.

### 5.3.2 Proposed Transition Strategy

The default QuadPlane transition strategy, while requiring tuning of some parameters, such as desired airspeed and servo tilt-angle through trial-and-error, is sufficiently generic to be used on a variety of hybrid UAV configurations. However, this default transition strategy requires some modifications before it can be adapted onto the AmphiQuad UAV.

One of the key factors to a successful transition to forward flight is the ability to quickly gain airspeed for the wing surfaces to produce lift. This can be done by tilting the motors forward either through the tilt-servos, or by pitching the nose of the aircraft downwards while being in the Quadrotor mode before the transition begins. The QuadPlane firmware does not allow incremental changes to the tilt-servo angle outside of the transition states. The all-or-nothing change in the tilt-servo angle results in a step increase of desired airspeed from zero to a user-defined value, while the thrust available from the motors to counteract gravity is being reduced by a factor of $\cos \gamma$. Although the motor-propeller combination was chosen to have sufficient thrust margin for controls, the need to maintain altitude and attitude at a large
tilt-servo angle can present thrust saturation problems. Moreover, the problem worsens as an increase in airspeed decreases the maximum dynamic thrust capable of being produced by the propellers.

A proposed workaround involves pitching the aircraft slightly downwards to allow it to accelerate forward and pick up some airspeed before activating the transition to forward flight. However, the default QuadPlane feature on ArduPlane does not automatically level the wing when the pilot commands a pitch-forward stick input. This results in a negative angle of attack of the lifting surfaces which leads to negative lift and more drag, and subsequently puts a higher demand on the propulsion system.

The different transition states mentioned in Section 5.3.1 involve the use of different controllers from both Quadrotor mode and Plane mode. During the TRANSITION_AIRSPEED_WAIT state, the attitude and altitude controllers from ArduCopter are used to stabilize the vehicle, while during the TRANSITION_TIMER state, only the attitude controller from the ArduCopter is used. Unless the user transitions into the Fly-By-Wire B fixed-wing flight mode, the aircraft will not attempt to maintain altitude after the transition is completed.

We want to implement a transition strategy where the altitude and attitude controllers are active throughout all transition states or at all tilt-servo angles. The controllers discussed in Chapter 4 are not dependent on the transition states of the UAV. In addition, we also want to allow the wing to level itself while a pitch forward command is given by the pilot such that the aircraft can gain some airspeed before the transition. In other words, the desired pitch angle commanded by the pilot will be converted into the angle of the tilt-servo to pitch the motors forward without affecting the attitude of the wing-body. Lastly, during transition, we want the angle of the tilt-servo (γ) to be commanded according to a Sigmoid function from γ = 0° (hover) to γ = 90° (forward flight) to ensure a gradual change the dynamics of the UAV.

5.4 List of Modifications Made to the Firmware

This section summarizes all the changes made to the ArduCopter and ArduPlane firmwares for the AmphiQuad UAV.

Support for Airspeed Sensor

Support for an airspeed sensor was originally only implemented in the ArduPlane firmware to improve altitude control [57]. As the proposed gain-scheduled altitude controller also requires airspeed data to calculate the error in the equilibrium speed, the codes related to the airspeed sensor were ported over from ArduPlane to ArduCopter.

New Flight Mode

The autonomous waypoint navigation (Auto) flight mode implemented in ArduCopter enables the UAV to visit the desired waypoints that were defined in a flight mission file prior
to operation with a user-defined desired ground speed. In this new **AutoPitch** flight mode, which was created based on the **Auto** flight mode, the UAV is to track a user-defined desired pitch angle instead of a desired ground speed. This allows the pilot to command the pitch angle of the quadrotor-body from 0° (hover) to -90° (forward flight) during transition.

**Tilt-Rotor Servo Controls**

Since the autopilot board is mounted on the quadrotor-body and pitches forward with the frame throughout transition, the wing has to be gimballed to remain level to generate lift as the airspeed increases. The **ArduCopter** firmware contains the camera gimbal feature which allows the user to use the tilt-servo for pitch stabilization of the wing-body. Modifications were made to allow setting an offset to the pitch angle of the quadrotor-body from the transmitter. For **ArduPlane**, modifications were made such that a commanded pitch angle change in VTOL mode is translated into a tilt-servo angle command.

**Gain-Scheduled Altitude Controller**

The gain-scheduled altitude controller presented in Section 4.1 was implemented on both the **ArduCopter** and **ArduPlane** firmwares. Five gain parameters were added for each 10 degree increment between -10 to -90 degree pitch angles of the quadrotor-body. The quaternion-based attitude controller was only used in the simulation but not implemented in the firmwares due to time constraints.

**Added Transition Trajectory**

The transition originally implemented within the **ArduPlane** QuadPlane feature involves the use of ramp-function in two stages as described in Section 5.3.1. Modifications were made to pitch the quadrotor-body frame continuously according to the Sigmoid function in Eq. (4.11).

**Summary**

This section outlined the two variations of ArduPilot firmwares that were studied and modified to implement the gain-scheduled altitude controller and the proposed transition strategy. The initial development and modifications on ArduCopter were done and tested on an experimental test frame. The challenges and difficulties faced during flight tests (that will be discussed in Section 7.1) prompted the change of focus towards developing the ArduPlane firmware, which was then used for the test flights on AmphiQuad UAV that will be discussed in Section 7.2.
Chapter 6

Simulation

This chapter presents the simulation model constructed in MATLAB Simulink that is used to model the forces and moments in Chapter 3 and to validate both the altitude controller and the attitude tracking controller in Chapter 4 during the transition from hover to forward flight.

6.1 MATLAB Simulink Model

The overall structure of the Simulink model is shown in Figure 6.1. The transition is achieved by commanding the pitch angle of the quadrotor frame to follow a trajectory shown in Eq. (4.11), with \( \theta_{Q,0} = 0^\circ \), \( \theta_{Q,f} = -90^\circ \), \( a = 0.25 \) and \( c = 50 \). The simulation was run using the ode4 (Runge-Kutta) solver with a fixed-time step of 0.01 seconds. This model, along with
the simulation results using the gain-scheduled altitude controller and quaternion-based attitude tracking controller, was presented at the 2018 AIAA Guidance, Navigation, and Control Conference [67].

As the attitude tracking control requires the angular velocity $\omega_{D}^{E}$ and angular acceleration $\dot{\omega}_{D}^{E}$ of the desired frame $\vec{F}_{D}$, those variables are calculated as follow,

$$\omega_{D}^{E} = 2\eta_{d}\dot{\varepsilon}_{d} - 2\dot{\eta}_{d}\varepsilon_{d} - 2\varepsilon_{d}^{\times}\varepsilon_{d}$$ (6.1)

$$\dot{\omega}_{D}^{E} = 2\eta_{d}\ddot{\varepsilon}_{d} - 2\ddot{\eta}_{d}\varepsilon_{d} - 2\varepsilon_{d}^{\times}\dot{\varepsilon}_{d}$$ (6.2)

where the derivatives of the quaternions for a rotation around the fixed axis $a = [0, 1, 0]^T$ are,

$$\dot{\varepsilon}_{d} = a\frac{1}{2}\dot{\theta}_{d}\cos\frac{\theta_{d}}{2}$$

$$\dot{\eta}_{d} = -\frac{1}{2}\dot{\theta}_{d}\sin\frac{\theta_{d}}{2}$$

$$\ddot{\varepsilon}_{d} = a\left(\frac{1}{2}\ddot{\theta}_{d}\cos\frac{\theta_{d}}{2} - \frac{1}{4}(\dot{\theta}_{d})^{2}\sin\frac{\theta_{d}}{2}\right)$$

$$\dot{\eta}_{d} = -\frac{1}{2}\ddot{\theta}_{d}\sin\frac{\theta_{d}}{2} - \frac{1}{4}(\dot{\theta}_{d})^{2}\cos\frac{\theta_{d}}{2}$$

With a Sigmoid function that is continuously differentiable for $\theta_{d}(t)$ in Eq. (4.11), we have,

$$\dot{\theta}_{d}(t) = a\theta_{d}(t)(1 - \theta_{d}(t))$$

$$\ddot{\theta}_{d}(t) = a^{2}\theta_{d}(t)(1 - \theta_{d}(t))(1 - 2\theta_{d}(t))$$

Apart from the UAV parameters summarized in Table 2.4, other parameters used in the simulation are summarized in Table 6.1. Assuming zero wind speed, the wing-body frame was kept at a constant pitch angle relative to the Earth to maintain a $8^\circ$ angle of attack during level flight, which provides the optimal speed for maximum range. The lift, drag and pitching moment coefficients used in the simulation were based on the aerodynamic analysis as discussed in Section 2.2, using the angle of attack $\alpha$ in the wing-body frame $\vec{F}_{B}$ calculated using Eq. (6.3), while the sideslip angle $\beta$ was assumed to be zero.

$$\alpha = \tan^{-1}\frac{w}{V}$$ (6.3)

The pole locations for determining the altitude controller gains were changed from -2 in Section 4.1.4 to -5 to achieve a faster response on the altitude and climb rate errors.

### 6.2 Simulation Results

**Quaternion-Based Attitude Controller + Gain Scheduled Altitude Controller (PP method)**

The simulation results, as shown in Figure 6.2, demonstrated that the UAV is capable of transitioning from hover to forward flight. Throughout the transition, the UAV was able to maintain
Table 6.1: Parameters used for simulation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive speed gain</td>
<td>$\sigma_1 = 2, \sigma_2 = 20$</td>
</tr>
<tr>
<td>Quaternion error gain, $k$</td>
<td>40</td>
</tr>
<tr>
<td>Filtered error gain, $K_1$</td>
<td>40.5 $1_{3\times3}$</td>
</tr>
<tr>
<td>Angular velocity error gain, $K_2$</td>
<td>53.3 $1_{3\times3}$</td>
</tr>
<tr>
<td>Integral gain, $k_3$</td>
<td>100</td>
</tr>
<tr>
<td>Elevon pitch moment, $C_{m,\delta_e}$</td>
<td>-0.027 /rad</td>
</tr>
<tr>
<td>Elevon lift slope, $C_{L,\delta_e}$</td>
<td>0.682 /rad</td>
</tr>
<tr>
<td>Elevon drag slope, $C_{D,\delta_e}$</td>
<td>0.1 $\cdot$sign($\delta_e$) /rad</td>
</tr>
<tr>
<td>Moment arm of motor</td>
<td>$l_{m,x} = 0.425$ m, $l_{m,y} = 0.410$ m</td>
</tr>
<tr>
<td>Motor Moment-to-Thrust ratio</td>
<td>$k_M = 0.1$</td>
</tr>
<tr>
<td>LQR altitude controller matrices</td>
<td>$Q = 0.25 \cdot 1_{3\times3}, R = 1 \cdot diag(1,1,5)$</td>
</tr>
</tbody>
</table>

...a stable altitude, with the final recorded change in altitude being 0.11 m. Despite the aerodynamic moments, the change in the position of the mass center throughout transition, and the gyroscopic couples, the attitude tracking controller was able to track the desired attitude to complete the transition.

The speed update equation was shown to be helpful in compensating for the deviation in the actual lift coefficient from the coefficient used for calculating the equilibrium speed due to the upward deflection of the elevons. The deflection reduces the lift generated by the wing, hence the UAV has to fly at a faster speed to maintain altitude.

It is also worth noting that motor 2 and 4 are operating at a much higher thrust while motor 1 and 3 are at idle (see Figure 3.3 for motor index). This is caused by the positive pitching moment due to the aft C.G. location in the wing (0.02 m behind the pivoting point, $h_W = 0.55\bar{c}$) and a negative elevon deflection. This could lead to control problems as the amount of total thrust required to overcome drag and amount of differential thrust required to overcome the pitching moment has to be within the performance limit of the propulsion system,

$$|M_d| \leq T_d \cdot l_{m,x} \quad \text{and} \quad 0 \leq T_d \leq T_{max}(v_{ax}) \quad (6.4)$$

With the new C.G. location being 0.1 m ahead of the pivoting point where the tilt-servo is mounted (at $h_W = 0.35\bar{c}$), the simulation results are shown in Figure 6.3. At the beginning of the transition, the mass center of the UAV was ahead of the origin of the reference frame, which creates a negative pitching moment and subsequently resulted in a higher thrust output in motor 1 and 3 (see Figure 3.3 for motor index). As the transition progressed, the downward pitch motion of the tilt-rotor frame moved the C.G. backwards, which resulted in a positive pitching moment and subsequently a higher thrust in motor 2 and 4.
Figure 6.2: Simulink Results Using Quaternion-Based Attitude Controller and Gain Scheduled Altitude Controller with PP Method, with Wing C.G. at $h_W = 0.55 \bar{c}$

Figure 6.3: Simulink Results Using Quaternion-Based Attitude Controller and Gain Scheduled Altitude Controller with PP Method, with Wing C.G. at $h_W = 0.35 \bar{c}$

Quaternion-Based Attitude Controller + Gain Scheduled Altitude Controller (LQR Method)

A similar simulation with the same gain scheduled altitude controller was run but with the gains determined using the LQR method instead of pole placement. The values of the $Q$ and $R$ matrices used to calculate the gain matrix $F$ are shown in Table 6.1. Other parameters used
in the simulation are identical to that of pole placement method in the previous section, with the C.G. in the wing located ahead of the pivoting point at $h_W = 0.35 \bar{c}$.

The simulation results are shown in Figure 6.4. The altitude controller with gains determined through the LQR method caused the UAV to lose 1.2 m of altitude throughout the transition, which is substantially larger compared to the pole placement method. The altitude loss resulted in a spike in thrust output and elevon downward deflection in an attempt to generate lift and upward force, which is a response to the slightly negative speed at around $t=5$ s. This could be solved by pitching the UAV forward slightly to gain some airspeed before activating the transition from hover to forward flight. Aside from the difference in altitude loss, the performance of the altitude controller with gains determined through the LQR method is similar to that with gains determined using pole placement.

![Figure 6.4: Simulink Results with Quaternion-Based Attitude Controller and Gain Scheduled Altitude Controller with LQR Method, with Wing C.G. at $h_W = 0.35 \bar{c}$](image)

**Remarks on the Gain-Scheduled Altitude Controller**

In these MATLAB Simulink simulations, the UAV was shown to be capable of maintaining the desired altitude and attitude throughout transition from hover to forward flight. In determining the gains of the gain-scheduled altitude controllers, the LQR method provided more flexibility in terms of weighing the control outputs and the errors on the state variables, but such it stills requires some tuning of the $Q$ and $R$ gain matrices to obtain satisfactory controller performance. As compared to pole placement, the LQR method assigns higher gains for the elevon outputs, which have led to earlier elevon actuations that caused oscillations in the control outputs. Nevertheless, both the LQR and pole placement methods provided a relatively
simple and automated way of determining the gains required for the altitude controller, which takes away a lot of the guesswork and trial-and-error to tune the individual gains at each increment of the scheduling variable. In this particular simulation scenario, the gains provided using pole placement performed better than those using the LQR method.

**Remarks on the Quaternion-Based Attitude Tracking Controller**

Throughout the simulations, the attitude tracking controller performed well in tracking the Sigmoid function of the desired pitch angle trajectory from 0° to -90°. The commanded motor torque is shown in Figure 6.5 for the simulation scenario using the gain-scheduled altitude controller with the gains determined using pole placement. There is a slight increase in the commanded torque at around 30 seconds which is in respond to the negative moment generated by the downward deflection of the elevons. The commanded torque changes sign due to the C.G. being shifted backwards as the quadrotor-body is tilted forward throughout the transition, showing that the quaternion-based attitude tracking controller is capable of handling the change in mass properties of the UAV.

![Figure 6.5: Commanded Torque in Simulation Using Quaternion-Based Attitude Controller](image-url)
Chapter 7

Experimental Work

This chapter describes the test flights conducted to validate the flight-worthiness of the UAV and the controllers proposed for the transition. A test frame was initially constructed to carry out incremental tests on the modifications made to the firmware, followed by flight tests on the AmphiQuad UAV. The technical challenges faced during the test flights are highlighted in this chapter, and some suggestions for future improvements are made.

7.1 Development and Flight Testing of Test Frame

7.1.1 Frame Design Overview

As the fabrication process of the AmphiQuad UAV is complex and time-consuming, any incident during flight tests could result in a lengthy repair process. As such, to minimize risks and ensure flight-readiness of the AmphiQuad UAV, a test frame shown in Figure 7.1 was constructed with the goal of experimenting with the autopilot firmware and making modifications to the source code. The test frame has an overall dimension of approximately 20” by 20”, with the carbon fiber tubes forming the quadrotor frame and the lifting surface being a

Figure 7.1: Design of The Test Frame UAV for Flight Testing

Not Shown: Carbon Fiber Landing Legs
low aspect-ratio micro aerial vehicle with a Bezier thin airfoil, which was inspired by Reference [68]. The test frame uses the same propulsion and avionics as the AmphiQuad UAV as detailed in Sections 2.6 and 2.8 respectively. The tilting mechanism is similar to the original design shown in Figure 2.11(a) in Section 2.4.

The main benefits of such a test frame are its simplicity to build, setup and repair with easily available commercial off the shelf (COTS) components, which results in shorter down time in case of any physical damage that occurred during test flights. This allows incremental flight tests to be carried out with every modification made to the firmware.

7.1.2 Flight Tests of Test Frame

Validation of Flight-Worthiness

A few test flights were performed to validate the flight-worthiness of the airframe. The test procedures for validating flight-worthiness of the UAV involved going through the firmware setup, performing preflight sensors (accelerometer and compass) and electronic speed controller (ESC) calibration on the ground, defining flight modes and failsafe actions, setting up the receiver and transmitter, as well as validating the motor spinning directions and connections to the autopilot board. After the completion of the ground test, the UAV was flown in the Stabilized mode to verify the performance and tune the rate controller, followed by Altitude Hold mode to verify the barometer reading and vibration damping performance of the autopilot board which could affect the performance of the altitude controller. Once the UAV was validated to fly with satisfactory performance in the manual stabilized mode, a waypoint navigation mission was set up for the UAV to navigate through under the Auto flight mode. Figure 7.2 shows the flight log one of the longer flights performed in Auto mode, with the maximum desired ground speed set to 10 m/s. The airspeed sensor readout, a feature custom-implemented in ArduCopter, was also tested in this flight.

New Flight Mode to Track Pitch Angle During Autonomous Mission

This test flight aimed to validate the new AutoPitch flight mode that tracks pitch angle (set to -12.5°) instead of the Auto flight mode that tracks a maximum desired ground speed (set to 2 m/s) in an autonomous waypoint mission. The UAV was set to take-off into Auto flight mode, and after positioning itself to have a sufficiently long airspace to fly in, the flight mode was switched into AutoPitch, which occurred around \( t=132 \) seconds in Figure 7.3.

Once the new flight mode was engaged, the pitch angle of the quadrotor-body closely tracked the desired pitch angle of -12.5°. The ground speed was no longer limited at a maximum of 2 m/s and the UAV slowly approached an equilibrium speed of 5 m/s. The default ArduCopter altitude controller was found to be capable of maintaining the altitude of the UAV at such pitch angle. A more extreme pitch angle was not tested due to the lack of sufficient airspace to conduct long distance flight, and the high ground speed gained through pitching
the UAV forward could lead to instability when switching from the AutoPitch to the Auto flight mode. A reverse transition strategy to gradually slow down the UAV is needed before a more negative pitch angle or a full transition to forward flight can be tested.

Modification to ArduCopter Gimbal Feature to Ensure Level Wing

The ArduCopter firmware was tested with the modifications made to the gimbal feature to ensure that the wing is automatically kept level as the quadrotor frame pitches forward. Figure 7.4 shows the portion of the test flight conducted under Auto flight mode, where the UAV varied its pitch angle to achieve the desired ground speed (to a maximum of 10 m/s) to visit
the waypoints pre-defined in the flight mission. As a result of the pitching motion, the PWM values of the tilt-servo increased to tilt the wing up relative to the quadrotor frame. Figure 7.4 shows the tilt-servo PWM output with pitch angle of the quadrotor frame.

![Figure 7.4](image)

**Figure 7.4:** Flight Test of Test Frame To Validate Functionality of Wing-Levelling

Gain-Scheduled Altitude Controller at Fixed Pitch Angle of Quadrotor-Body

The UAV was initially set to take-off into Auto flight mode, then the flight mode was switched into AutoPitch through the transmitter, which occurred around $t=58$ seconds in Figure 7.5. The initial desired pitch angle was set to be 0° and then to -15° according to the Sigmoid function in Eq. (4.11). The final pitch angle was set to be only -15° to prevent the UAV from gathering too much speed and fly beyond the boundary of the flying field. The gain-scheduled altitude controller was used in the AutoPitch flight mode, with the control gains and parameters being identical to those used in the simulation in Section 6.1.

As observed in the figure, the altitude and pitch angle of the UAV started oscillating widely, which can be traced to the aggressive throttle output which shows only either 0% or 100% values after the gain-scheduled controller was activated. The flight mode was hence switched back to Auto and the UAV was safely landed.

![Figure 7.5](image)

**Figure 7.5:** Flight Test of Test Frame To Activate Gain-Scheduled Altitude Controller

The output of the gain-scheduled altitude controller described in Section 4.1 is a commanded thrust output, which can be translated into a commanded acceleration in the $z$-axis of the quadrotor-body frame. The motors, however, require PWM values that correlate to the an-
gular speed of the propellers. The ArduCopter implementation uses an acceleration to throttle function which uses the acceleration measured from the Inertial Measurement Unit (IMU) as feedback for a PID controller to output the desired throttle. To ensure that the PID controller was not the cause of the oscillatory behaviour, a look-up table was implemented to determine the throttle level based on airspeed and desired thrust as described in Section 2.6. While the throttle look-up table method was tested to work well in simulation, the discrepancy between the propulsion data provided by the analysis tool and the real-world performance caused a similar oscillatory behaviour to be observed in the subsequent flights.

Another plausible cause for the oscillations in the throttle outputs is the unsuitable reference inputs for the altitude controller. Upon switching into the AutoPitch flight mode, the gain-scheduled altitude controller commanded a zero horizontal speed at zero pitch angle, which the UAV was almost able to achieve at $t=60$ seconds. The wind condition on the day of flight may have resulted in a drift in the velocity of the UAV that caused the altitude controller to aggressively compensate for, which subsequently triggered the oscillatory behaviour in the throttle output and hence the altitude. One possible workaround is to initialize a non-zero pitch angle and horizontal speed to reduce the drifting effects from the wind. This hypothesis was not put to test due to weather and time constraints.

### 7.1.3 Challenges in ArduCopter Firmware Implementation

Apart from the oscillatory behaviour observed in the throttle output when using the gain-scheduled altitude controller, the main hindrance to the continued development to the AutoPitch flight mode is the lack of plane-related navigation control in the ArduCopter firmware. When using the AutoPitch flight mode, the UAV tends to drift sideways and deviate from its desired path in the perpendicular direction of the vector from the previous waypoint to the next waypoint, which leads to an increasing cross-track error as illustrated in Figure 7.6. This is because the flight mode was originally set up to track a desired pitch angle, while keeping a zero roll angle and a zero yaw rate. The zero roll angle and zero yaw rate should keep the UAV on the track between the waypoints, but wind conditions and asymmetricity in mass distribution caused the UAV to drift from the desired trajectory.

Modifications were made to the firmware such that the attitude controller will receive roll inputs based on the cross-track error from the navigational controller. However, the modifications resulted in oscillatory roll and yaw behaviours as the UAV was approaching the next waypoint, as shown in Figure 7.7. Due to time constraints and limited scope of the research, further studies into the implementation details of the ArduCopter navigational controller were not explored. A potential workaround is to define the next waypoint very far away from the previous waypoint where the transition begins, but it leads to safety concerns of UAV breaching the prescribed flight boundary or a “fly-away” UAV in case of communication lost over long distances.
Figure 7.6: Cross-Track Error and Distance to Waypoint in the **Auto/AutoPitch** Flight Mode

Figure 7.7: Divergence in Navigational Output Roll and Yaw Angles in **AutoPitch** Flight Mode Approaching Next Waypoint

An alternative method was then employed; this method involves switching from the **AutoPitch** mode to the **Auto** mode when the UAV is in close proximity to the next waypoint. As a result of this workaround, the pitch angle of the UAV cannot be set to a large negative value or the UAV will pick up significant amount of speed. This may subsequently cause a loss of attitude stabilization when the UAV has to abruptly slow down upon switching from the **AutoPitch** mode to the **Auto** mode. This is observed in Figure 7.8 where the UAV pitched up violently and subsequently lost altitude to recover its attitude after slowing down.

Figure 7.8: Aggressive Roll/Pitch Maneuvers When Switching from **AutoPitch** to **Auto** Mode

While the test frame is not a scaled version of the AmphiQuad UAV and does not share the same mass and aerodynamic properties, it has provided valuable flight operation experi-
ence and has helped familiarization with the working details of the autopilot firmware. The final design of the AmphiQuad UAV described in Chapter 2 was influenced by the lessons learned from the test frame, which includes but not limited to the addition of flexible carbon fiber landing gears, the improved design of the servo tilting mechanism, the revised motor mount design, and the revised avionics mounting location. More importantly, it showed that a substantial amount of development work is required on the ArduCopter firmware before a successful transition can be demonstrated.

### 7.2 Flight Testing of AmphiQuad UAV

#### 7.2.1 Flight in Quadrotor (Hover) Mode

Multiple test flights were conducted purely in the quadrotor flight modes to ensure the flight-worthiness of the airframe and the functionality of the autopilot. Unlike the test frame UAV which was operated using the ArduCopter firmware with the Pixhawk autopilot board mounted on the quadrotor-body, the AmphiQuad UAV used the ArduPlane firmware with the Pixhawk autopilot board mounted within the wing-body.

These test flights have led to several design changes as detailed in Chapter 2, such as a reinforced interface between the carbon fiber tubes on the quadrotor frame, a revised tilt-rotor interface that is more secured and reliable in operations, and a longer servo actuation arms to increase deflection of the control surfaces. These test flights also resulted in a significant overhaul of the electrical system within the UAV, as the high current draw by the propulsion system resulted in overheated wires due to the internal resistance of the cables that were insufficiently thick. This resistance resulted in the energy loss in the form of heat dissipation, which restricted the power available to the motors and caused the UAV to be operated near maximum throttle. As shown in Figure 7.9(a) and 7.9(c), the throttle level slowly drifted above 80% and the UAV started to lose altitude. The 16 AWG wires were replaced with 12 AWG and the XT60 wire connectors are replaced with XT90 wire connectors. As shown in Figure 7.10(c), subsequent flights showed substantial improvement in throttle performance, where the throttle remained around 70% during hover flight.

The quadrotor attitude controller was tuned through trial-and-error to improve the responsiveness of the UAV in roll and yaw. The tuning was done by providing sinusoidal inputs through the transmitter while flying the UAV in the QStabilize mode. This tuning was necessary as the large lifting surfaces in the $xz$ and $xy$-plane caused a weathercock effect that induced undesirable roll and yaw motions that were difficult to be countered by the differential thrusts. The insufficient yaw control authority on the UAV was further compounded by the structural limitations of the quadrotor-body, as the low torsional stiffness of the quadrotor frame caused the motor arms to deflect in the direction that generated an opposing moment to the desired yaw motion. This led to undamped yaw oscillations that were difficult to tune, as shown in Figure 7.10(f). As the motors were selected based on the preliminary mass budget
Figure 7.9: VTOL Flight of AmphiQuad UAV on June 28, 2018
Figure 7.10: VTOL Flight of AmphiQuad UAV on July 3, 2018
of 4.5 kg, the thrust margin available for control was lower than originally designed as the final take-off mass of the UAV eventually became 5.45 kg. Both weathercock effects and a lower thrust margin for control led to saturation of motor outputs which made precise hovering and heading hold difficult in the quadrotor hovering mode.

### 7.2.2 Transition from Hover to Forward Flight

The test flight for transitioning from hover to forward flight was conducted on July 5, 2018 at the University of Toronto’s Koffler Scientific Research (KSR) at Jokers Hill in King City, ON. The purpose is to validate the flight-worthiness of the UAV using the default transition strategy implemented on the QuadPlane as detailed in Section 5.3.1, which will act as a baseline for evaluating the transition performance before validating the custom controllers on the autopilot firmware. The UAV took off vertically in the quadrotor stabilized (QStabilize) mode to gain sufficient altitude before activating the transition into the fixed-wing stabilize (Stabilize) mode. The flight data are plotted in Figure 7.11.

When the transition was activated, the autopilot tilted the motors forward and went into a “wait for airspeed” mode while the differential thrusts produced by the motor continued to stabilize the UAV. When the transition began, the UAV initially experienced some pitch oscillations which were quickly dampened. However, once the UAV reached an airspeed of 12.5 m/s (ground speed of 8.4 m/s), the pitch diverged – the UAV pitched up slightly and then started a nose down maneuver. Efforts were made to recover the UAV by switching back to the QStabilize mode, but the UAV did not have sufficient altitude for recovery. The video of the flight is available on [https://www.youtube.com/watch?v=lEjRM692QnA](https://www.youtube.com/watch?v=lEjRM692QnA).

One possible cause of the crash was determined to be caused by the longitudinal instability of the UAV as it transitioned into the fixed-wing forward flying mode. The transition maneuvers resulted in the C.G. of the UAV being shifted backwards aft of the neutral point. As the airspeed increases over the lifting surfaces, the effects of the longitudinal instability magnifies and the pitch angle of the UAV diverges. While there is no official documentation on the ArduPlane website [57], it was believed that the default attitude controller implemented was capable of controlling a longitudinally unstable aircraft. There was no extensive experimental study on using the ArduPilot firmware on small-scale unstable aircraft, with the exception of Reference [69] which suggested that a slow response of elevator servo can affect pitch damping performance, and that the D gain on the pitch rate controller should be increased significantly (by a factor of 3 or more) for aircraft with negative static margin. This pitch instability was not discovered during the simulation as the simulation used a different attitude controller from the controller implemented in ArduPlane. Due to time constraints, the ArduPlane attitude controller was not replicated on the simulator, and the quaternion-based attitude controller presented in Section 4.2 was not implemented on the ArduPlane firmware.

On the other hand, just before the pitch divergence occurred, motor 3 (front left) was at the lowest throttle while motor 4 (aft right) was at the highest throttle setting. This suggests that
Figure 7.11: Transition Flight of AmphiQuad UAV on July 5, 2018
the pitching moment due to the aft C.G. position might have been too large for the differential thrusts to counteract, as the dynamic thrust available decreases with increased airspeed in the axial direction of the spinning propellers. As shown in Figures 7.11(e) and 7.11(f), the lack of differential thrust was worsened by the correction of the oscillating yaw before the transition started, with motor 3 and 4 outputs being saturated. This reduced the differential thrust margin available for pitch control, which consequently reduced the maximum pitching moment that can be generated by the motors. This lack of differential thrust available was not discovered during the simulation in Chapter 6 as the C.G. was assumed to be closer to the leading edge. During the test flight, the C.G. placement in the wing was subjected to space constraints of a flying-wing body design, which required the batteries to be placed further backwards than they were originally intended. The simulation was also unable to capture the yaw control difficulties due to the low torsional stiffness of the quadrotor frame that led to a reduction in the differential thrust available for pitch control during the transition flight.

7.3 Suggestions for Improvements

This section presents several key improvements that can help preventing a similar flight incident on July 5, 2018 from reoccurring during the transition from hover to forward flight, as well as suggesting design changes that can improve the flight-testing process of the UAV.

Variable-Pitch Propellers

Variable-pitch propellers are increasingly used in quadrotors to increase the agility of the vehicle, increase propeller efficiency and flight endurance, as well as overcoming the actuator bandwidth limitations of fixed-pitch propellers [70, 71]. In a hybrid UAV like the AmphiQuad UAV, the propeller choice presented in Section 2.6 was mainly driven by the dynamic thrust available at the cruising airspeed, in which a higher airspeed would require a higher pitch propeller. However, such propeller choice was a design compromise, as a low-pitch propeller for multirotors hovering flight is more efficient and draws less current for the same amount of thrust produced. At the cost of increasing mechanical complexity, having variable-pitch propellers on the AmphiQuad UAV will help to solve these trade-offs between the two distinct flight regimes of hover and forward flight, as well as increasing control bandwidth through manipulating the pitch of the propellers to generate larger differential thrust during high speed flight.

Due to time and resource constraints, the use of variable-pitch propellers was not explored. The ArduPilot firmware currently only supports variable-pitch propellers in the ArduCopter firmware through the HeliQuad vehicle, where all four motors are commanded to spin at the same rotational speed while the four individual servos control the pitch of each propeller [57]. Modifying the ArduPlane firmware to enable such feature on the QuadPlane module would take considerable amount of time and effort, and is hence left as part of the future work.
Increase Torsional Stiffness of Quadrotor Frame

The current quadrotor frame on the AmphiQuad UAV was made out of four commercial off-the-shelf (COTS) carbon fiber tubes as shown in Section 2.3. While these tubes are cheap and easily available, the small cross sectional area formed by joining them together led to poor torsional stiffness which contributed to the yaw oscillations seen in Section 7.2.

A revised design such as shown in Figure 7.12 could help improve the torsional stiffness of the quadrotor frame. However, such a design needs to be lightweight and custom-sized to fit the design of the UAV, which can only be achieved through composite lay-ups within a custom designed mold. This revised design was out of the scope of the current thesis.

Mass Redistribution to Prevent Shift in Center of Gravity

As the ability of the ArduPlane firmware to stabilize longitudinally unstable aircraft is unclear, in addition to the difficulties in tuning and test flying an unstable aircraft, the AmphiQuad UAV should be designed with a positive longitudinal static margin. In the initial stage of the design, the tilt-axis was chosen to be in the mid-chord position of the wing to satisfy the dimension constraints and to prevent interference between the propellers and the wing. Further design revisions can be made to optimize the location of the tilt-axis to move the overall C.G. forward. In addition, further design considerations showed that it might have been possible to mount the batteries on the quadrotor frame through an unconventional extension of the carbon fiber frame as shown in Figure 7.13, with the extension length depending on the weight of the batteries and the remaining weight of the quadrotor frame.

Implementation of Quaternion-Based Attitude Controller

An alternative to the proposed design change to prevent the shift in C.G. during transition (Figure 7.13) is to implement the quaternion-based attitude controller presented in Section 4.2 to replace or to complement the default attitude controller implemented on the ArduPlane firmware. While the quaternion-based attitude controller was simulated in MATLAB Simulink
in Section 6.1 for the transition from hover to forward flight, it was not implemented in the ArduPlane QuadPlane module due to time constraints. Future studies and more sophisticated modelling of the actuator delays may also be necessary to fine tune the control gains, as actuator delays can affect pitch damping in an unstable aircraft [69].

**Design Changes to Enable Take-Off in Fixed-Wing Mode**

The main challenge in test flying the AmphiQuad UAV was the inability to take off in a fixed-wing configuration. This is largely due to the front propellers, which when fully tilted forward in a fixed-wing configuration, will strike the ground during take-off. The UAV is too heavy for a hand-launch, and was not designed with specific mounts or any structural consideration to take-off on a launcher. Moreover, due to the dimension of the quadrotor frame, a long nose landing gear would be required to provide sufficient ground clearance, but it also has to be carefully sized to prevent physical interference with the quadrotor frame in the hovering mode.

The capability of taking off and landing in fixed-wing configuration will provide the opportunity to validate the flight worthiness, performance, and stability of the UAV independent of the quadrotor hovering mode. This would also allow the actuators to be trimmed and the controller gains to be further tuned based on the flight performance. This would eliminate any uncertainties or issues pertaining to the fixed-wing flying mode before activating the transition from hover to forward flight.
Chapter 8

Conclusions

This research involves the process of designing, controlling, and operating a hybrid UAV that combines the benefits of both fixed-wing aircraft and multirotor UAVs for the purpose of a wetland algal boom inspection mission. In summary, this thesis first discussed the existing hybrid UAV configurations in the literature, along with the transition controls and strategies employed. A tilt-rotor configuration was selected because it is more stable than a tail-sitter and less mechanically complex than a tilt-wing configuration, while still being able to be designed for water flotation capability. The design of the AmphiQuad tilt-rotor UAV was presented in detailed in regards to the aerodynamic design, propulsion analysis and selection, the servo tilting-mechanism, the structural design, avionics selection, and the construction techniques and material selection. The performance of the UAV was analyzed with a take-off mass of 5.45 kg, and was found to provide a maximum endurance of 30 minutes or a maximum range of 21.5 km in fixed-wing forward flying mode.

The thesis also developed a dynamic model of the UAV by considering all the forces and moments through the built-up method. The attitude and altitude controllers implemented on the ArduCopter and ArduPlane variants of the ArduPilot firmware, as well as the existing transition strategy implemented on the QuadPlane module were studied and discussed. A transition strategy with continuous change in the pitch angle of the quadrotor-body was proposed with a gain-scheduled altitude controller and a quaternion-based attitude controller, and the controllers were validated through simulation in MATLAB Simulink to show successful transition from hover to forward flight.

In the process of implementing the gain-scheduled altitude controller on the autopilot board of the AmphiQuad UAV, an easy-to-build test frame was constructed to incrementally test the modifications made to the firmware. The development of firmware was initially done on the ArduCopter variant of the ArduPilot firmware, but the effort was discontinued due to time constraints and technical challenges faced in terms of discrepancies in throttle performance, as well as difficulties in integrating navigation control and flight controls of a fixed-wing aircraft. The development was continued on the ArduPlane variant of the firmware to conduct test flights on the AmphiQuad UAV using the QuadPlane module. The Amphi-
Quad UAV was successfully flown in quadrotor hovering mode after several design iterations were made and repeated tuning of the attitude controllers. Unfortunately, during a transition attempt from hover to forward flight, the AmphiQuad UAV experienced a brief moment of pitch divergence and crashed. The reasons for the crash were identified to be a combination of lack of pitch control bandwidth based on the differential thrusts on the motor, as well as the lack of longitudinal stability that was not accounted for by the autopilot attitude control as the C.G. of the UAV being shifted backwards during transition. A list of suggestions of design changes and improvements was made to ensure successful transition flight of the AmphiQuad UAV and reliable flight operation for the wetland inspection mission in the future.

While the proposed design remained a proof-of-concept prototype and the flight test results showed limited success, the results presented in this thesis highlighted the challenges associated with designing hybrid fixed-wing VTOL UAVs, while outlining the necessary design, control and practical implementation considerations, in addition to the software and tools required as part of the process for validating the design and flight performance. This will allow future research to build upon the results and the suggestions for improvement from this thesis. Future development and research in this field will eventually enable more UAVs with long endurance and precise hovering capabilities, which will be useful in various applications that include but not limited to environmental monitoring, parcel delivery, humanitarian aids delivery, and forest fire monitoring.
Appendix A

Online Resources and Repositories

A.1 AVL Files

The input files that define the geometry of the wing and the vertical tails can be found at the online repository https://bitbucket.org/utiasfsc/avl_files/src.

![Figure A.1: Geometry of the Lifting Surfaces Defined in AVL](image)

(a) Wing-Body  
(b) Vertical Tail

A.2 ArduPilot Firmware

The top-level code repository for the ardupilot firmwares can be found at https://bitbucket.org/utiasfsc/ardupilot_mod/src/, with the original open-sourced code at Ref. [65].

ArduPlane

This thesis referenced the codes in the ArduPlane version 3.8, which can be found at the specific branch URL of https://github.com/ArduPilot/ardupilot/blob/plane3.8/. The modified code implementing the changes described in Section 5.4 can be found at https://bitbucket.org/utiasfsc/ardupilot_mod/src/Plane_AmphQuad. The codes referenced in this thesis are summarized in Table A.1.
Table A.1: Files and Codes Related to ArduPlane Firmware

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArduPlane/Attitude.cpp</td>
<td>Scaling on control surfaces output based on airspeed</td>
</tr>
<tr>
<td>ArduPlane/tiltrotor.cpp</td>
<td>Tilt-rotor specific codes, including servo-tilting mechanism and motor outputs</td>
</tr>
<tr>
<td>ArduPlane/quadplane.cpp</td>
<td>Hybrid UAV (tilt-rotor, tail-sitter) codes, including transition states and controller switching</td>
</tr>
<tr>
<td>libraries/AP_TECS/AP_TECS.cpp</td>
<td>Altitude controller based on Total Energy Control System</td>
</tr>
<tr>
<td>libraries/APM_Control/</td>
<td></td>
</tr>
<tr>
<td>AP_PitchController.cpp</td>
<td>PID controller on pitch rate, P control on pitch angle error</td>
</tr>
<tr>
<td>AP_RollController.cpp</td>
<td>PID controller on roll rate, P control on roll angle error</td>
</tr>
</tbody>
</table>

ArduCopter

This thesis referenced the codes in the ArduCopter version 3.6, which can be found at the specific branch URL of https://github.com/ArduPilot/ardupilot/blob/Copter-3.6/. The modified code implementing the changes described in Section 5.4 can be found at https://bitbucket.org/utiasfsc/ardupilot_mod/src/Copter_AmphiQuad. The codes referenced in this thesis are summarized in Table A.2.

Table A.2: Files and Codes Related to ArduCopter Firmware

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>libraries/AC_AttitudeControl/</td>
<td></td>
</tr>
<tr>
<td>AC_PosControl.cpp</td>
<td>PID controller on z-acceleration, P control on z-velocity, and sqrt controller on altitude error</td>
</tr>
<tr>
<td>AC_AttitudeControl.cpp</td>
<td>PID controller on body angular rates, sqrt controller on attitude error, calculate quaternion error</td>
</tr>
<tr>
<td>AC_AttitudeControl_Multi.cpp</td>
<td>Output commanded torques to individual motors on multirotor through differential thrusts</td>
</tr>
</tbody>
</table>

A.3 ArduPilot Documentation

The documentation on the ArduPilot firmware can be found in Ref [57], with the following pages used throughout this thesis:

- ArduPlane – QuadPlane Parameter setup: /plane/docs/quadplane-parameters.html
- ArduPlane – QuadPlane Support: /plane/docs/quadplane-support.html
- ArduPlane – Flight Modes: /plane/docs/flight-modes.html
- ArduCopter – HeliQuads (Variable Pitch Multicopters): /copter/docs/heliquads.html
- ArduCopter – Altitude Hold Mode: /copter/docs/altholdmode.html
- ArduCopter – Connect ESCs and Motors: /copter/docs/connect-escs-and-motors.html
Appendix B

Aerodynamics Coefficients

This appendix shows the body-axes aerodynamic coefficients for the wing-body and quadrotor-body (vertical tail) as obtained from the AVL analysis.

Table B.1: Reference Dimensions of Lifting Surfaces

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Main Wing</th>
<th>Vertical Tail</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference span</td>
<td>$b_{wing} = 2.0$</td>
<td>$b_{VT} = 0.5$</td>
<td>m</td>
</tr>
<tr>
<td>Reference area</td>
<td>$S_{wing} = 0.882$</td>
<td>$S_{VT} = 0.144$</td>
<td>m$^2$</td>
</tr>
<tr>
<td>Reference chord</td>
<td>$\bar{c}_{wing} = 0.473$</td>
<td>$\bar{c}_{VT} = 0.288$</td>
<td>m</td>
</tr>
</tbody>
</table>

Table B.2: Aerodynamic Coefficients of Lifting Surfaces in Body-Axes

| $\alpha$ (deg) | Wing-Body | | | Vertical Tail | | |
|-----------------|-----------|-----------|-----------|---------------|-----------|
|                 | $C_x$     | $C_z$     | $C_m$     | $\beta$ (deg) | $C_y$ | $C_l$ | $C_n$ |
| -9              | -0.7311   | 0.7833    | 0.1789     | 0             | 0.00000  | 0.00000  | 0.00000 |
| -7              | -0.5223   | 0.5946    | 0.1482     | 5             | -0.00077 | 0.00000  | 0.00012 |
| -5              | -0.3464   | 0.4239    | 0.1204     | 10            | -0.00153 | 0.00000  | 0.00023 |
| -3              | -0.2065   | 0.2675    | 0.0950     | 15            | -0.00228 | 0.00000  | 0.00035 |
| -1              | -0.1050   | 0.1217    | 0.0714     | 20            | -0.00301 | 0.00000  | 0.00046 |
| 1               | -0.0434   | -0.0178   | 0.0486     | 25            | -0.00372 | 0.00000  | 0.00056 |
| 3               | -0.0213   | -0.1551   | 0.0262     | 30            | -0.00440 | 0.00000  | 0.00067 |
| 5               | -0.0080   | -0.2918   | 0.0037     | 35            | -0.00504 | 0.00000  | 0.00077 |
| 7               | 0.0116    | -0.4269   | -0.0187    | 40            | -0.00565 | 0.00000  | 0.00086 |
| 9               | 0.0374    | -0.5599   | -0.0408    | 45            | -0.00622 | 0.00000  | 0.00094 |
| 11              | 0.0687    | -0.6902   | -0.0626    |               |         |         |        |
| 13              | 0.1034    | -0.8177   | -0.0842    |               |         |         |        |
| 15              | 0.1212    | -0.9477   | -0.1066    |               |         |         |        |

$C_m$ output is at 25% mean aerodynamic chord ($\bar{c}$) position on the chord line. $C_l$ and $C_n$ outputs are at 0% position on the chord line. The results with negative $\beta$ angles are the negative of the results from the positive $\beta$ angles. Note that the results presented in this table represent the coefficients of one of the two vertical tails on the UAV. The results have to be multiplied by 2 to account for both vertical tails.
Bibliography


