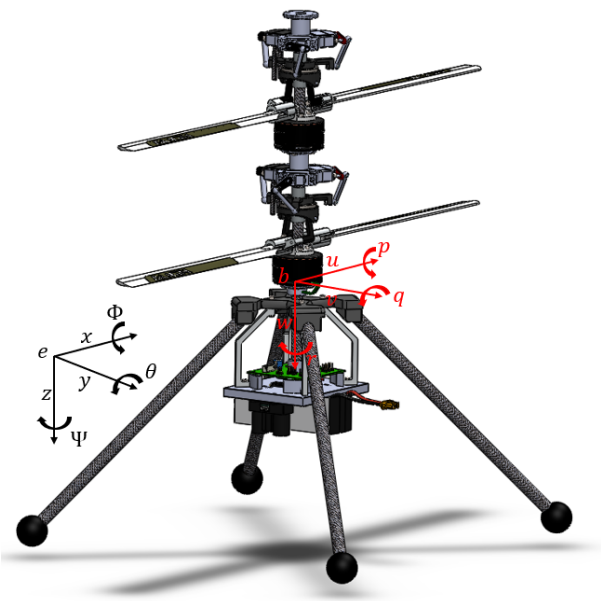
IMPERIAL

Imperial College of Science, Technology and Medicine

Department of Mechanical Engineering

Design, Modelling, and Control of an Unmanned Coaxial Helicopter

Final Report



Author:
Jansen Papworth

Project Supervisor:
Dr. Ludovic Renson

Project co-Supervisor:
Dr Hamed Rezaee

Page Count: 40
June 6, 2024

Abstract

As a species, we are going to Mars. Before we get to Mars, it is desirable to accurately map out large portions of terrain, and aerial vehicles are able to do this significantly faster than rovers. However, flying on mars is difficult because of the very low atmospheric density; hence, very large blades are needed to sustain flight. It is often considered that a coaxial helicopter is the best architecture to achieve this. A coaxial helicopter consists of a single shaft with two rotors rotating in two different directions to balance the torque. Controlling a coaxial helicopter introduces significant complications due to the cross-coupling introduced by the two rotors rotating in close proximity. In this report, the controls of such a helicopter are studied. Firstly, a coaxial helicopter was designed for flight on earth. The equations of motion governing the flight of this helicopter were derived. These equations of motion were then implemented into a computational model in order to design a robust controller and study the dynamics of the system. Flight software was written capable of detecting faults and automatically aborting tests, ensuring safety for the hardware and the operator. The manufactured helicopter was tested in an incremental process by enabling each controller axes independently, then building up to a full flight test. The full flight test was successfully achieved and the helicopter was able to hover stably for 3.5 seconds. However, additional gain tuning from the simulated model would likely result in an improved response. Gain tuning on the flying test article is dangerous, and hence additional model refinement and gain tuning should be completed prior to additional flight testing.

Acknowledgements

Firstly I would like to express my gratitude to Dr. Ludovic Renson for choosing to supervise this project after originally supervising the design, manufacture and testing portion of the mars helicopter project in a previous year.

I would also like to thank Dr. Hamed Rezaee for providing help in constructing the linear model and choosing the necessary controller input variables.

My thanks to Dr. Maria J. D. Ribera Vicent for her expertise in helicopter models and systems and providing me with the much-needed starting papers and textbooks when undertaking this project.

Finally, I would like to thank Dr. Mihailo Ristic for answering my numerous questions on control systems related to this coaxial helicopter.

Contents

1 Introduction and Coaxial Helicopter Overview **1**
1.1 History of Coaxial Helicopter Development . . . 2
1.2 Coaxial Helicopter Design . . . 3
1.3 Flight Controls Circuit Board . . . 5
1.4 Manufactured Helicopter . . . 5
1.5 Helicopter Properties . . . 5
1.6 Swash Plate Geometry . . . 7

2 System Model and Dynamics **8**
2.1 General Equations of Motion . . . 9
2.1.1 Helicopter's Force Vectors . . . 9
2.1.2 Blade Flapping Angles . . . 10
2.1.3 Rotor Torque Vectors . . . 11
2.1.4 Thrust and Torque Expressions . . . 12
2.1.5 Thrust and Torque Coefficient Derivations . . . 12
2.2 Required States and Sensors . . . 13
2.3 Coordinate Transforms . . . 14
2.4 Swash Plate Kinematics . . . 14
2.4.1 Kinematics Mathematical Derivations . . . 15
2.4.2 Kinematics Unit Testing . . . 16

3 Controls Design **17**
3.1 Plant Linearisation and Poles . . . 17
3.2 Full Closed Loop Controller . . . 18
3.2.1 Controls Mixing Block . . . 19
3.2.2 Yaw Mixer . . . 20
3.3 Model Gain Tuning and Discussion . . . 21
3.3.1 Pitch and Roll Controller . . . 21
3.3.2 Heave Controller . . . 22
3.3.3 Yaw Controller . . . 22
3.3.4 Position Controller . . . 23
3.4 Closed Loop Gains and Discrete Poles . . . 24

4 Flight Software **25**
4.1 Software State Machine . . . 25
4.2 Magnetometer Calibration . . . 26
4.3 Discrete Controller Software Implementation . . . 27

5 Flight Testing Results and Discussion **28**
5.1 Yaw Testing Stand Tests . . . 28
5.2 Suspended Yaw Controller Testing . . . 29
5.3 Heave Controller Testing . . . 30
5.4 Pitch, Roll and Position Controllers . . . 31
5.5 Full Flight Testing . . . 31

6 Conclusion **34**

Nomenclature

β_{ci}	Cos flapping angle
β_{si}	Sin flapping angle
$\boldsymbol{\omega}_b$	Body angular velocities
$\boldsymbol{f}_{b,g}$	Gravity force in body frame
$\boldsymbol{f}_{b,R_i}$	Rotor force in body frame
$\boldsymbol{r}_{b,R_i}$	Rotor position vector
$\boldsymbol{t}_{b,Ri}$	Rotor torques in body frame
$\boldsymbol{v}_b$	Body velocities
$\boldsymbol{y}$	Output Vector
λ	Lock number
λ_i	Inflow ratio
$\mathbf{M}_{col}$	Collective mixing
$\mathbf{M}_{cyl,l}$	Lower cyclic mixing
$\mathbf{M}_{cyl,u}$	Upper cyclic mixing
$\mathbf{M}_{rpm}$	Rotational mixing
$\mathbf{u}$	Input vector
$\mathbf{x}$	State vector
ω_i	Rotational speed
ϕ, θ, ψ	Euler angles
ϕ_i	Servo actuator angles
ϕ_{roll}	Commanded roll
ψ_{yaw}	Commanded Yaw
σ	Rotor solidity
θ_{ci}, θ_{si}	Swash plate angles
θ_{h,R_i}	Heave angle
θ_{pitch}	Commanded pitch
ζ	Damping ratio
A	State matrix
A_b	Blade area
A_d	Disk area
B	Input matrix
C	Output matrix
c	Chord length
C_Q	Torque coefficient
C_T	Thrust coefficient
e_k	Discrete error
H_r	Neutral swash height
$h_{z,i}$	Rotor hub distance
I	Inertial matrix
k	Discrete time index
k_C	Critical gain
K_d	Derivative gain
K_i	Integral gain
K_p	Proportional gain
L	Servo arm length
$l_{A_1B_1}$	Servo arm to swash distance
L_{bc}	Non-rotating arm lengths
m	Helicopter mass
N	Number of blades
p, q, r	Body Angular rates
Q	Torque
R	Radius rotor
r	Swash plate radius
T	Sampling time
T_r	Rotor thrust
u, v, w	Body Velocities
u_k	Discrete input
x, y, z	Inertial positions
z_{heave}	Commanded heave

1 Introduction and Coaxial Helicopter Overview

When looking to fly in the Martian atmosphere, several considerations have to be made to determine the best system architecture. There are traditional well-studied options for earth such as quad-copters, standard helicopters, or tandem helicopters. The advantages of these are discussed with respect to martian flight conditions in [1] where it was concluded that because of the low atmospheric density a coaxial helicopter was the best architecture. This followed a similar conclusion NASA made for the mars helicopter technology demonstrator in [2]. A coaxial helicopter that was developed for Mars flight and was preliminary tested by a DMT project and is shown in Figure 1. A coaxial helicopter consists of two counter rotating rotors balancing torque, instead of the more conventional tail rotor found on traditional helicopters.

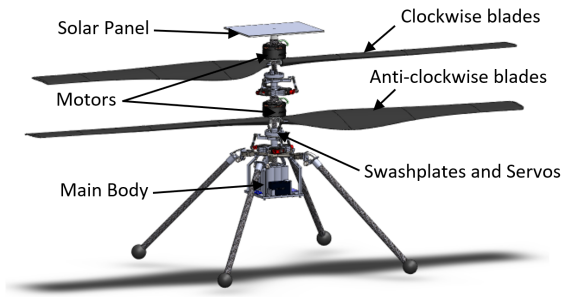


Figure 1: *Coaxial helicopter designed for martian operation in [1].*

In this project, the control systems required to fly this helicopter were developed. Before undertaking the controller design and testing, the helicopter was re-designed for earth operation. This involved incorporating the recommended design improvements detailed in [1]. Additionally, the martian blades were replaced with standard earth helicopter blades. Then, new blade grips were designed and manufactured for these Earth blades. A flight controller PCB was designed and manufactured that could seamlessly integrate all the sensors and outputs required to fly the helicopter. After the manufacturing was completed, the helicopter systems were tested independently prior to test flights. The hardware designed for the coaxial helicopter has each rotor set having a swash plate. The swash plate allows for the variation of the angle of attack for each of the blades. Controlling this variation in angle of attack allows for the full six-degree-of-freedom control which is required for flight.

The objectives of this project are to demonstrate successful control in Earth operation, achieving sustained stable flight with full closed-loop feedback. To achieve this a detailed model of the helicopter's equations of motion were derived. These equations were then implemented in

a MATLAB Simulink model. In this model, the controllers were designed and tuned. The final controller consisted of decoupling the helicopter states and passing them back through a nested PID controller. This PID controller then outputs the pitch, roll, yaw, and heave commands. From these controller commands, inverse kinematics was derived to calculate the underlying positions of the servo actuators. Finally, the controller board was programmed to the designed controller specification, with additional time spent ensuring the helicopter could communicate with a ground station over Bluetooth. Finally, flight tests were conducted, where the results were compared to the helicopter model and conclusions were drawn.

1.1 History of Coaxial Helicopter Development

Only five years after the first powered flight in 1903 some of the earliest development attempts of Vertical Take-Off and Landing (VTOL) aircraft are documented, as shown in [3]. Initial attempts of these VTOL aircraft were unsuccessful due to lack of engine power, but most earlier designed VTOL aircraft did attempt to utilise the two-rotor coaxial design. Much later in 1953 the earliest successful helicopters began to enter the market, spearheaded by Igor Sikorsky, who was the pioneer of the earlier unsuccessful coaxial helicopter attempts. These helicopters adopted a swash plate design to control the pitch and roll. A swash plate mechanism was used as shown in Figure 2 (right), which has three pitch links attached to a non-rotating plate and two pitch links attached to the rotors on the rotating plate.

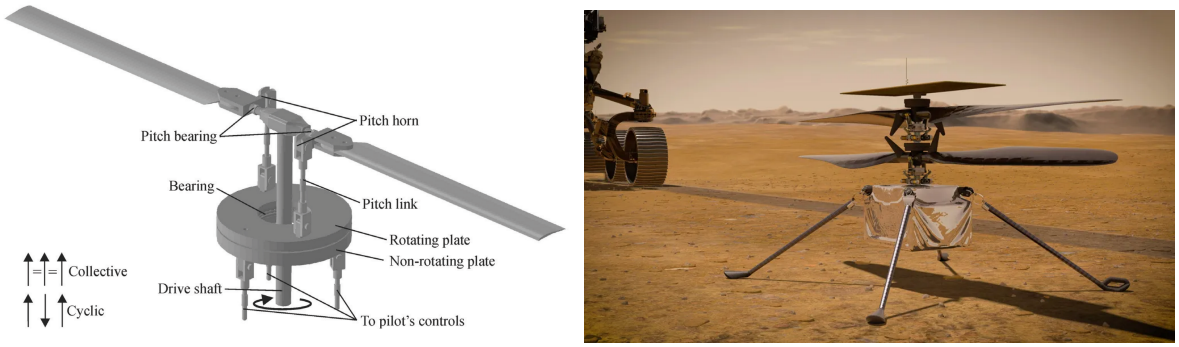


Figure 2: *(left) Helicopter Swash Plate from [4], (right) Ingenuity, Nasa's coaxial helicopter technology demonstrator [5].*

The non-rotating plate is actuated by the three links creating an angle on the rotating plate. This angle is called the cyclic input and is the effective position steering input to a helicopter.

This cyclic angle allows for this steering by pointing the rotors thrust vector. Additionally, the entire swash plate can be moved up and down to increase and decrease the lift in a collective motion. A combination of cyclic and collective inputs allows for total 6 degree-of-freedom control of the helicopter.

Igor Sikorsky was of the opinion that one rotor with torque being counteracted by a small tail fan would result in a more successful design. He was proved correct with today around 95% of helicopters being this single rotor design [3]. However, when flying in the low density Mars atmosphere coaxial designs allow for large rotors necessary to provide the required lift. In 2020 Nasa and JPL demonstrated a coaxial helicopter design on Mars successfully with Ingenuity, as seen in Figure 2 (right) [5]. Control of these coaxial helicopters introduces several complications both with the design and with the controls, and it is desirable to understand this for future coaxial flight development.

1.2 Coaxial Helicopter Design

The original designed helicopter shown in Figure 1 was redesigned for Earth operation. This redesigned helicopter is shown in Figure 3 (left) and included updating the swashplates geometry, blade grips, blades, main body, battery mount, sensor mounting, motor mounting, shaft, shaft hubs and manufacturing a flight controller PCB. Additionally, the rotors and swash plates orientation was reversed meaning only the upper motor wires and 18 smaller servo wires had to pass through the hollow shaft compared to all motor wires and 9 servo wires on the previous helicopter. This change allowed for a thicker shaft to be used, reducing the risk of torsional failure.

The blades were then replaced with standard Earth helicopter blades which were selected from a range of options in [6]. The chosen rotors comprise of a E-139 aerofoil and are commonly used on the trex-450, a small-scale remote control helicopter [7]. These rotors were especially ideal for the coaxial helicopter, as they are perfectly symmetrical, preventing the need for a different anticlockwise rotor.

The new rotors required new blade grips; the blade grips are the component that connects individual rotors to the main rotor hub. This grip needs to be able to rotate about the axis

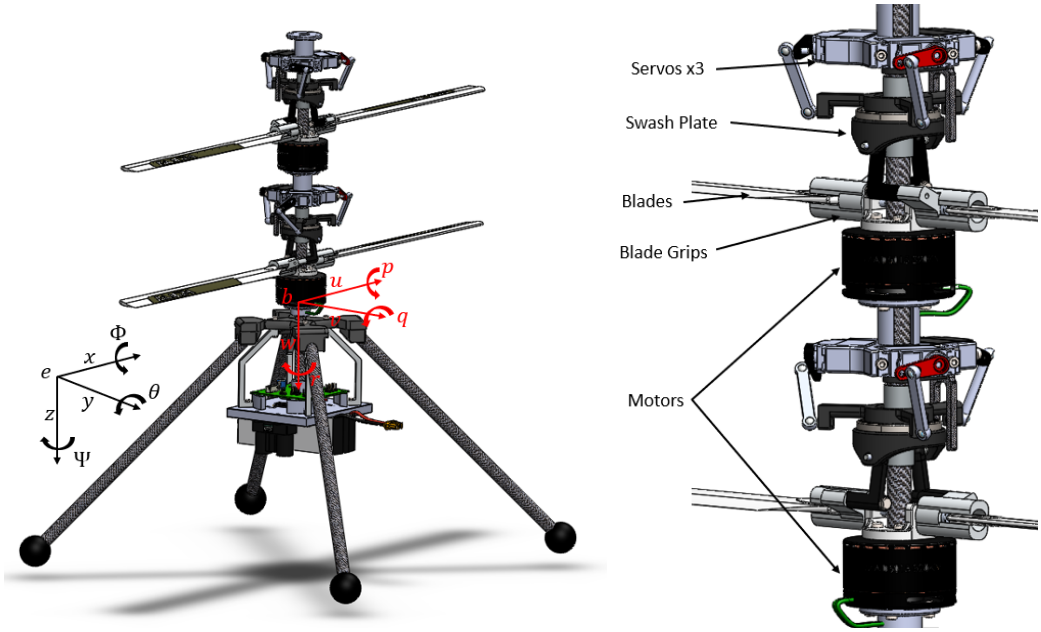


Figure 3: *(left) Designed Earth helicopter and defined coordinate system, (right) labelled helicopter hardware.*

perpendicular to the hub to allow for variation of the angle of attack of the rotor. All whilst sustaining the centripetal force generated by the spinning rotors. The design used a three bearing setup, with two ball bearings to support the bending moment and an additional thrust bearing to support the centripetal load. The detailed explanation of the design is shown in the Appendix A. These parts were machined out of aluminium and undesirably a clockwise and anticlockwise grip were required due to the swash plate geometry.

Finally, the motor hubs (the component that mounts the motors to the shaft) were redesigned to incorporate the servo mounting plate. This change prevents any torsional stress from passing through the shaft section between these two components. This is desirable as this is where there is a slot in the shaft for all of the control wires to pass through. A detailed CAD image of the redesigned helicopter is shown in Figure 3 (right). Here, all the controllable hardware is visible, with there being eight inputs to this system. These inputs are upper and lower motor rotational speed (2 inputs), lower swash plate servo angles (3 inputs), upper swash plate servo angles (3 inputs).

This hardware is controlled by sending pulse width modulated signals which correspond to a specified speed or angle. For the motors the controllable range is $800rpm - 1500rpm$ while

servos can be controlled from $0^0 - 180^0$. However, the servo ranges is limited in the software to prevent hardware damage.

1.3 Flight Controls Circuit Board

Due to the custom nature of the designed helicopter, there are no commercially available flight controller circuit boards that met all of the requirements. Hence, it was necessary to design and manufacture a custom board. This board had to power all the servos, motor speed controllers, and data sensors while communicating with a ground station. The power requirements were complicated with two required voltages, 3.3 volts for the flight computer, IMU and LIDAR, whilst 5 volts was required for the servos and motors controllers. In addition a $660\mu F$ capacitor was placed over the LIDAR sensors power as required by the data sheet. The final circuit board used header pins to connect the data and power wires to the servos, LIDAR sensor, and motor controllers. Whilst incorporating all other sensors directly on the board. The designed flight controllers PCB engineering drawings are shown in Appendix B.

1.4 Manufactured Helicopter

The final manufactured helicopter is shown in Figure 4 with all the hardware integrated. This manufactured helicopter went through a series of hardware tests to ensure that data was being received from all onboard sensors correctly. In addition, each servo was swung throughout its entire operating range and the motors were spun independently. This test verified successful control of the hardware and the hardware development element of this project was paused.

Figure 4: *Manufactured coaxial helicopter.*

1.5 Helicopter Properties

Before modelling helicopter flight dynamics, several fundamental quantities need to be established from the designed and built helicopter. These quantities are listed in Table 1. The

mass was simply determined directly by weighing the built hardware. Additionally, the mass of the motors, batteries, and speed controllers were independently found. This mass data along with the entire helicopter mass was entered into the helicopters CAD model. This software could then accurately determine the helicopter centre of gravity and the inertias.

The inertias are calculated about the centre of gravity and formed a non-diagonal inertial matrix. However, as the diagonal terms dominate it is assumed to be diagonal as given in:

$$I = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{zz} & 0 \\ 0 & 0 & I_{yy} \end{pmatrix}. \tag{1}$$

This assumption is appropriate as the non-diagonal terms are an order of magnitude lower than the diagonal terms. This assumption simplifies the equations of motion by removing the cross-coupling introduced by a non-diagonal inertial matrix. The final parameters in Table 1 are the non-dimensional helicopter properties. The lock number is the ratio of aerodynamic forces to inertial forces of the rotor blades. These estimates assume that the blades used in this helicopter are very light and very stiff [8]. Rotor solidity is the ratio of the area of the rotor blades A_b to the area of the rotor disk A_d [9] as given in:

Table 1: *Helicopter constants.*

Property	Value (Units)
g	9.81 (m/s^2)
ρ (Air density)	1.225 (kg/m^3)
h_{z1}	0.06717 (m)
h_{z2}	0.02212 (m)
m	2.112 (kg)
R	0.3775 (m)
c (Chord length)	0.03 (m)
I_{xx}	59.3×10^{-6} (kg·m^2)
I_{yy}	55.6×10^{-6} (kg·m^2)
I_{zz}	19.6×10^{-6} (kg·m^2)
λ (Lock Number)	15
σ (Rotor Solidity)	0.054

$$\sigma \equiv \frac{A_b}{A_d} = \frac{2cR}{\pi R^2} = \frac{2c}{\pi R}. \tag{2}$$

Both rotor solidity and Lock number have significant effect on the performance of the helicopter's controller and as such may be considered design parameters when attempting to improve the helicopter flight dynamics and control systems.

1.6 Swash Plate Geometry

A close-up section of the designed swash plate assembly is shown in Figure 5 (left). The non-rotating part of the swash plate is visible along with the rotating part. The servo on the left of this image is designated servo 1 with its corresponding angle being ϕ_1, the next two servo angles being designated ϕ_2 and ϕ_3 respectively. The swash plate is comprised of a stationary sub-module and a rotating sub-module as shown in Figure 5 (centre). Finally, the swash plate coordinate system is defined in Figure 5 (right). There is a system defined at the base of each servo denoted by A_i, a system defined at the end of each servo arms input rocket arm, C_i, and another defined at the end of the lower rod B_i.

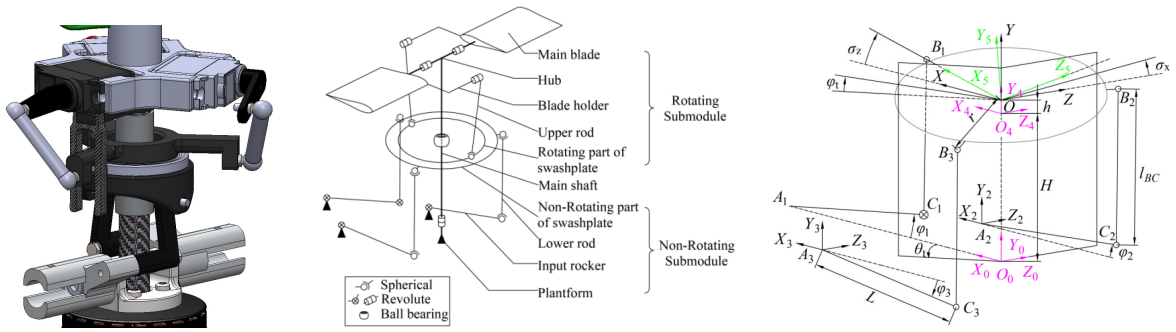


Figure 5: *(left) Designed swash plate assembly, (centre) swash plate joints [10], (right) swash plate coordinate systems [10].*

The dimensions of the designed swash plate are defined in Table 2. All of these dimensions are design parameters that can be adjusted to improve the kinematics of the swash plate. In general, it is desirable for the swash plate to be "geared up", meaning that large servo movements are required to create relatively small swash plate plane angle variations. This performance is desirable as it results in very precise commanded swash plate

Table 2: *Swash Plate Geometry*

Constant	Value [cm]
H_r (neutral swash plate height)	3.5
L_{bc} (non-rotating arm lengths)	3.5
L (servo arm length)	2.5
r (swash plate radius)	4.3
A_{angles} (angle between each arm)	120°

plane angles[1]. The coordinates of the servo horns base A in the neutral base frame A^0 is defined in:

$$A^0 = \begin{pmatrix} A_1^0 \\ A_2^0 \\ A_3^0 \end{pmatrix} = \begin{pmatrix} a_{x1}^0 & a_{x2}^0 & a_{x3}^0 \\ a_{y1}^0 & a_{y2}^0 & a_{y3}^0 \\ a_{z1}^0 & a_{z2}^0 & a_{z3}^0 \end{pmatrix} = \begin{pmatrix} 4.9 & 0 & -0.8 \\ -3.9 & 0 & 3.2 \\ -3.9 & 0 & -3.2 \end{pmatrix} \text{ [cm]}, \tag{3}$$

where A_1^0 refers to the vector position of the first servo in base coordinate system. As the servos horns location is fixed with respect to this base coordinate system all parameters are constants. Finally, the coordinates of B in the fifth coordinate system B^5 are defined as:

$$B^5 = \begin{pmatrix} B_1^5 \\ B_2^5 \\ B_3^5 \end{pmatrix} = \begin{pmatrix} b_{x1}^5 & b_{x2}^5 & b_{x3}^5 \\ b_{y1}^5 & b_{y2}^5 & b_{y3}^5 \\ b_{z1}^5 & b_{z2}^5 & b_{z3}^5 \end{pmatrix} = \begin{pmatrix} r & 0 & 0 \\ -r\sin(30) & 0 & r\cos(30) \\ -r\sin(30) & 0 & -r\cos(30) \end{pmatrix} \text{ [cm]}, \tag{4}$$

where C_1^0 refers to vector position of the first swash plate servo arm connector. As the fifth coordinate system is centred about the base of the swash plate these values are all constants.

2 System Model and Dynamics

Prior to designing a robust controller, a strong model of the underlying system is required. This model is found by deriving the underlying equations of motion of the helicopter and follows a similar approach to that taken in [11]. These underlying equations are significantly simplified with the following assumptions.

1. The aerodynamic rotor forces of each rotor are independent to each other.
2. The principle axis of inertia are parallel to the body frame coordinate system.
3. Helicopter motion is slow compared to rotor motion.
4. Rotor thrust vectors are perpendicular to the rotor tip path plane.
5. Rotor can be considered hinge less (rotors are rigidly connected to the grips [12]).

[1]This is an area to improve this helicopter's design, currently there is only 15° of swing in the servos allowed and it is desirable have around 45°.

2.1 General Equations of Motion

Firstly, the helicopter coordinate system was defined in Figure 3 (left). Here, the reference coordinate system for the body and inertial frame is defined with the standard notation [13]. Standard notation defines vertical height downwards counter to intuition; this was used to maintain continuity with the right-hand coordinate system and literature in this field. The Euler angles, pitch θ, roll ϕ and yaw ψ are defined as the orientation between the Earth and body frame coordinate systems.

The general equations of motion for a system with six degrees of freedom are given in [13]. These can be applied to the coaxial helicopter including summed notation for the two rotor forces and are given by:

$$\dot{\boldsymbol{v}}_b = \frac{1}{m}\left(\sum_{i=1}^{2} \boldsymbol{f}_{b,R_i} + \boldsymbol{f}_{b,g}\right) - \boldsymbol{\omega}_b \times \boldsymbol{v}_b, \quad \dot{\boldsymbol{\omega}}_b = I^{-1}\left(\sum_{i=1}^{2} \boldsymbol{t}_{b,R_i} - \boldsymbol{\omega}_b \times (I \cdot \boldsymbol{\omega}_b)\right), \tag{5}$$

where the body velocities are $\boldsymbol{v}_b = (u \quad v \quad w)^\top$ and the angular rates are $\boldsymbol{\omega}_b = (p \quad q \quad r)^\top$. The rotor force in the body frame is $\boldsymbol{f}_{b,R_i}$, whilst the rotor torque in the body frame is $\boldsymbol{t}_{b,R_i}$ and the gravity force in the body frame is $\boldsymbol{f}_{b,g}$. These equations have the fully diagonal inertial matrix assumption applied removing the cross coupling introduced between the angular accelerations.

2.1.1 Helicopter's Force Vectors

The gravitational force vector always acts directly down in the inertial coordinate system (i.e in z), hence to find this force in the body coordinate system it must be rotated by the roll and pitch Euler angles:

$$\boldsymbol{f}_{b,g} = R_1(\phi) \cdot R_2(\theta) \cdot \begin{pmatrix} 0 & 0 & m \cdot g \end{pmatrix}^\top. \tag{6}$$

where the transformation matrices are defined in the standard form as:

$$R_1(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c(\alpha) & -s(\alpha) \\ 0 & s(\alpha) & c(\alpha) \end{pmatrix}, R_2(\beta) = \begin{pmatrix} c(\beta) & 0 & s(\beta) \\ 0 & 1 & 0 \\ -s(\beta) & 0 & c(\beta) \end{pmatrix}, R_3(\gamma) = \begin{pmatrix} c(\gamma) & -s(\gamma) & 0 \\ s(\gamma) & c(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{7}$$

where c represents cos and s represents sin. The rotor frame is defined by the plane of the tip path of the rotors themselves and is centered at the base of the rotor hub. The forces generated in this rotor frame $\boldsymbol{f}_{r,R_i}$ are given by:

$$\boldsymbol{f}_{r,R_i} = \begin{pmatrix} 0 & 0 & -T(\theta_{h,R_i}, \omega_i) \end{pmatrix}^\top, \tag{8}$$

where T is the rotor thrust and is a function of heave angle θ_{h,R_i} and rotational speed ω_i. This rotor force is perpendicular to the rotor frame. This rotor frame is defined by the angles that the tip path of the rotors creates from the body frame. These are called flapping angles and are shown in Figure 6. These angles $\beta_{c,i}$ and $\beta_{s,i}$ are used to find the rotor force in the body frame, with the rotation matrices given in Equation (7):

$$\boldsymbol{f}_{b,R_1} = R_2(\beta_{c1}) \cdot R_1(-\beta_{c1}) \cdot \boldsymbol{f}_{r,R_1}, \quad \boldsymbol{f}_{b,R_2} = R_2(\beta_{c2}) \cdot R_1(\beta_{s2}) \cdot \boldsymbol{f}_{r,R_2}, \tag{9}$$

2.1.2 Blade Flapping Angles

To understand the angle the rotor force vector creates, it is necessary to describe these rotor flapping angles. Rotor flapping is a critical element of helicopter flight dynamics and occurs when the rotor blades oscillate upward and downward as they rotate around the hub, as shown in Figure 6 [14]. In this figure, both the longitudinal flapping β_{cosine} and the lateral flapping β_{sin} angles are illustrated together with the corresponding coining angle with respect to the body frame of the helicopter. As the coining angle does not affect the tip path plane, it is of little concern for the control system analysis[2].

The flapping process itself forms a highly coupled dynamical system coupled to all the helicopter states. Detailed analysis of this dynamical rotor flapping is found in [12]. However, for the purpose of this model a simplified first-order approximation will be used, this simplification follows previous work in [8] and [11]. It is shown in these works that when rigid rotor blades are assumed the flapping coefficients can be written as linear combinations of the helicopter velocity states as given by:

[2]Coining angle would be studied in detail for the stress analysis of rotors.

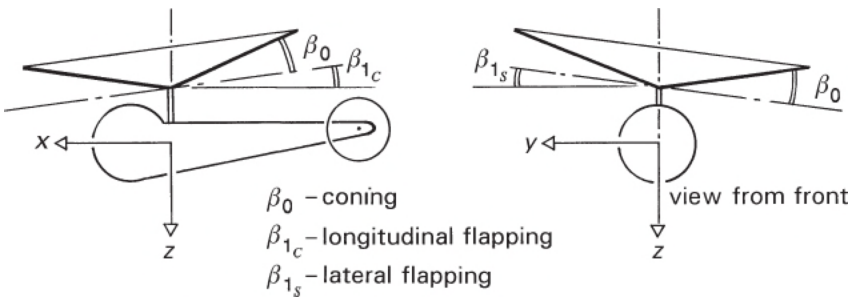


Figure 6: *Blade flapping angles [14].*

$$\beta_{c1} = \theta_{c1} + \beta_{c1,u} \cdot u + \beta_{c1,q} \cdot q + \beta_{c1,p} \cdot p, \quad \beta_{s1} = \theta_{s1} + \beta_{s1,v} \cdot v + \beta_{s1,p} \cdot p + \beta_{s1,q} \cdot q, \tag{10}$$

$$\beta_{c2} = \theta_{c2} + \beta_{c2,u} \cdot u + \beta_{c2,q} \cdot q + \beta_{c2,p} \cdot p, \quad \beta_{s2} = \theta_{s2} + \beta_{s2,v} \cdot v + \beta_{s2,p} \cdot p + \beta_{s2,q} \cdot q, \tag{11}$$

where flapping coefficient derivatives are denoted by $\beta_{ci,u}$, which is the linear contribution of the velocity u to the cosine flapping angle. These coefficients are calculated from the helicopter parameters detailed in Table 1 and the equations are shown in Appendix C. The input angles θ_{c1}, θ_{s1}, θ_{c2}, θ_{s2} are the swash plate plane angles, which are controlled by commanding the swash plates servo actuators. Controlling these angles allows the controller to point the rotors thrust vector into the desired location. More complicated analysis of the flapping angles [8] would include further helicopter states and introduce their non linear contributions. However for a small scale helicopter the difference should be minimal as the assumption of rigid rotors is very accurate.

2.1.3 Rotor Torque Vectors

Next the body torques on the helicopter are considered such that the angular rates in Equation (5) can be found. The rotor torques in the body frame $\boldsymbol{t}_{b,R_i}$ are given by:

$$\boldsymbol{t}_{b,R_i} = \boldsymbol{r}_{b,R_i} \times \boldsymbol{f}_{b,R_i} + \boldsymbol{t}_{b,D_i}, \tag{12}$$

where $\boldsymbol{r}_{b,R_i}$ is the position of the rotor hub from the centre of gravity given by $\boldsymbol{r}_{b,R_i} = (0 \;\; 0 \;\; -h_{z,i})^\top$, where $h_{z,i}$ is the height of the rotor hub from the centre of gravity. The underlying drag

vector $\boldsymbol{t}_{b,D_i}$ in the body frame is given:

$$\boldsymbol{t}_{b,D_i} = \begin{pmatrix} 0 & 0 & (-1)^i Q(\theta_{h,R_i}, \omega_i) \end{pmatrix}^\top, \tag{13}$$

where Q is the rotors drag force and is a function of heave angle θ_{h,R_i} and rotational speed ω_i in the same way as the thrust. For a coaxial helicopter, each rotor spins in different directions balancing the torque, hence the negative sign on the first anticlockwise spinning motor.

2.1.4 Thrust and Torque Expressions

Finally to close the equations of motion, the expressions for thrust T and torque Q are found [12]. These expressions are given in:

$$T(\theta_h, \omega) = C_Q(\theta_h)\rho\pi R^4\omega^2, \quad Q(\theta_h, \omega) = C_T(\theta_h)\rho\pi R^5\omega^2, \tag{14}$$

where the C_T is the thrust coefficient and C_Q is the torque coefficient both of which are a functions of the heave angle and rotor rotational speed.

2.1.5 Thrust and Torque Coefficient Derivations

Traditional blade element theory would directly integrate the lift coefficient C_L over the distance of the blade to derive the thrust coefficient. However as documented in [12] and shown in Figure 7 (left), the root of the blade and the tip introduce significant losses in lift from this ideal.

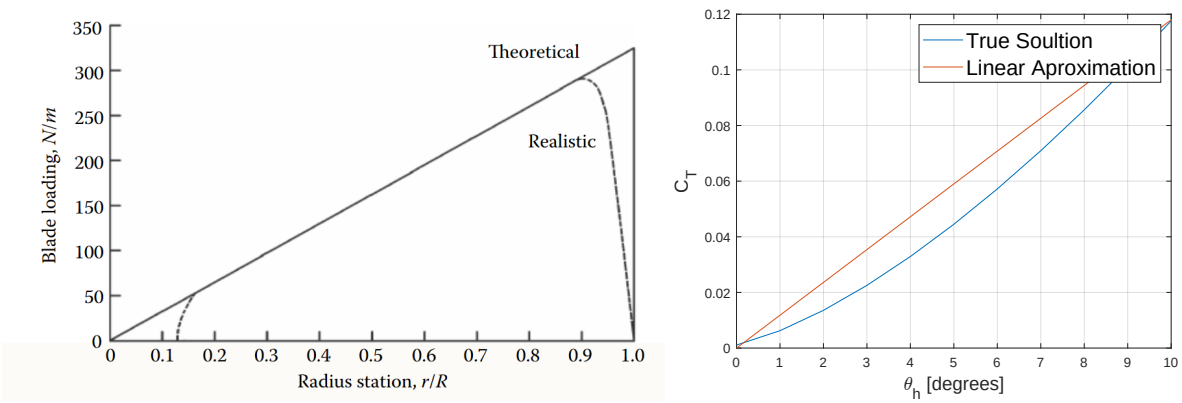


Figure 7: *(left) Blade loading against radius demonstrating the loss of lift present in the tip and the root [12], (right) derived thrust coefficient against angle of attack.*

Prandtl gave an expression for this tip loss factor as a function of the number of blades and the thrust coefficient in [15]. This derivation involves numerically solving for the inflow ratio λ_i and is shown in Appendix D, along with the Matlab code used to numerically solve this equation [16] [17]. After the inflow ratio is found this is integrated across the radius of the blade in:

$$C_T = \int_0^R 4\lambda_i^2 r dr, \tag{15}$$

to find the thrust coefficient. This inflow ratio and thrust coefficient calculation is then performed from zero to ten degrees angle of attack and is shown in Figure 7 (right). It can be seen that the true solution is non linear, however in an effort to maintain simplicity in the model and the designed controller, this curve was linearised to $C_T = 0.0118 \cdot \theta_h$. A more sophisticated model may want to incorporate a higher order interpolation to better represent this function. After finding the thrust coefficient, the torque coefficient can be found by applying the relationship $C_Q = C_T\sqrt{\frac{C_T}{2}}$ [12].

2.2 Required States and Sensors

The equations of motion have now been closed, and these equations are seen as functions of eight input helicopter variables. These input variables can be summarised in the input matrix $\boldsymbol{u}$ as given by:

$$\boldsymbol{u} = \begin{pmatrix} \theta_{h1} & \theta_{c1} & \theta_{s1} & \theta_{h2} & \theta_{c2} & \theta_{s2} & \omega_1 & \omega_2 \end{pmatrix}^\top. \tag{16}$$

These input variables are what the controller can change to stabilise the helicopter. The helicopter state vector $\boldsymbol{x}$ is then given in:

$$\mathbf{x} = \begin{pmatrix} x & y & z & u & v & w & \phi & \theta & \psi & p & q & r \end{pmatrix}^\top. \tag{17}$$

The state vector comprises of 12 states containing the inertial frames positions (x, y, z), body frame velocities (u, v, w), the Euler angles (ϕ, θ, ψ) and the body frame angular velocities (p, q, r).

To control the helicopter accurately, it is desirable to measure as many of these as possible. However, with the designed flight controller board, it was unable to directly measure all of

these states. Angular rates are measured directly through the gyroscope, Euler angles can be found using an orientation filter combining data from the accelerometer, gyroscope and magnetometer [18], and LIDAR can read the inertial frame height z directly. However, this still leaves states x, y, u, v, w without being read directly. The velocity w is calculated by taking the derivative of z. Unfortunately, the only way to find x, y, u, v is to integrate and rotate the accelerometer readings. This significantly amplifies noise and is subject to drift over time, hence a more sophisticated helicopter should incorporate a method of measuring these position states directly. As the helicopter's ultimate goal is to fly on Mars where GPS is not available, other methods are required to find these states. [2] discuses using a camera onboard and horizontal LIDAR sensors but this was considered beyond the scope of this project.

2.3 Coordinate Transforms

The equations of motion are expressed in the body frame, however, the position in the inertial frame is what is of concern. Therefore, there is a need to transform from the body frame to the inertial frame. This involves rotating the body states by the pitch, roll and yaw, to go to the equivalent inertial states, as shown in:

$$\begin{pmatrix} \dot{x} & \dot{y} & \dot{z} \end{pmatrix}^\top = R_3(-\psi) \cdot R_2(-\theta) \cdot R_1(-\phi) \cdot \begin{pmatrix} u & v & w \end{pmatrix}^\top . \tag{18}$$

Then the rate of rotation of the Euler angles is found by rotating the body velocity vector into the inertial frame as shown in:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{pmatrix} \cdot \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{19}$$

2.4 Swash Plate Kinematics

The input matrix given by Equation (16) contains the angles of the swash plate $\theta_h, \theta_c, \theta_s$, however, the actual hardware shown in Figure 5 has three stationary servos. These three servos must be commanded in such a way as to generate the desired swash plate angles. This process

is referred to as inverse kinematics and can be seen in Figure 8, which shows the process of moving from the original commanded swash plate angles to the underlying servo actuator angles ϕ_1, ϕ_2, ϕ_2.

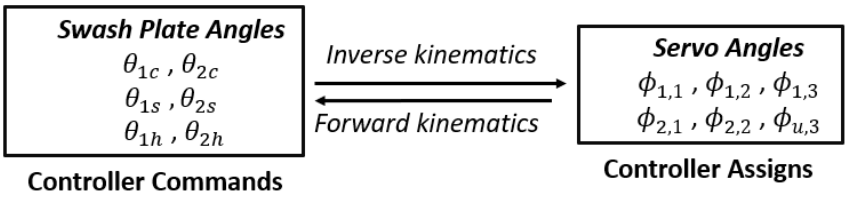


Figure 8: *Forward and inverse kinematics process.*

The flight controller must perform this inverse kinematic calculation in real time, hence it is desirable to keep the equations light. To do this the matrices have been expanded and the expanded expression are implemented directly into the flight software.

2.4.1 Kinematics Mathematical Derivations

The kinematic derivation follows a similar approach to that taken in [10] and [19]. With the critical dimensions used detailed in Table 2. The derivation is similar to that of a standard planar manipulator. As such, the first step is to determine the location of swash plate in the zeroth coordinate system, B^0 which is given in:

$$B^0 = B^5 \cdot C_5^0(\theta_h, \theta_c, \theta_s) = \begin{pmatrix} b_{x1}^0 & b_{x2}^0 & b_{x3}^5 \\ b_{y1}^0 & b_{y2}^0 & b_{y3}^5 \\ b_{z1}^0 & b_{z2}^0 & b_{z3}^5 \end{pmatrix}, \tag{20}$$

where the transformation matrix C_5^0 is a function of $\theta_h, \theta_c, \theta_s$ and its derivation is shown in Appendix E. The dimension matrix B^5 is defined in Equation (4) and is used to find ϕ_1:

$$l_{A_1B_1} = \sqrt{(b_{x1}^0 - a_{x1}^0)^2 + (b_{y1}^0 - a_{y1}^0)^2 + (b_{z1}^0 - a_{z1}^0)^2}, \tag{21}$$

$$\phi_1 = \sin^{-1}\left(\frac{b_{y1}^0}{l_{A_1B_1}}\right) - \cos^{-1}\left(\frac{L^2 + l_{A_1B_1}^2 - L_{bc}^2}{2L \cdot l_{A_1B_1}}\right). \tag{22}$$

With all the swash plate constants being defined in Table 2. These equations form a triangle between the base of the servo horn A and the ball joint of the swash plate B. Then they calculate the angle required to form this triangle. This is performed for the other two servo actuators, and due to the symmetrical geometry of the swash plate, only one more equation is required as given in:

$$\phi_i = \sin^{-1}\left(\frac{b^0_{xi}}{\sqrt{(b^0_{xi})^2 + (b^0_{yi})^2}}\right) - \sin^{-1}\left(\frac{L^2_{bc} - L^2 - (b^0_{xi})^2 - (b^0_{yi})^2 - (b^0_{zi})^2}{2L\sqrt{(b^0_{xi})^2 + (b^0_{yi})^2 + (b^0_{zi})^2}}\right), \tag{23}$$

where the index i takes values 1, 2 for each servo respectively. A more detailed description of the flight software implementation of these equations is shown in Appendix E.

2.4.2 Kinematics Unit Testing

After the kinematics was derived the equations were tested in Python. This testing involved varying each input parameter $\theta_h, \theta_c, \theta_s$ individually and then varying them all together to ensure that no discontinuities are exhibited in the operating range of the swash plate. If discontinuities are present in the required operating range, the design of the swash plate would need to be altered, as this would indicate a plane angle that is not assignable. The tested swash plate graphs are shown in Figure 9.

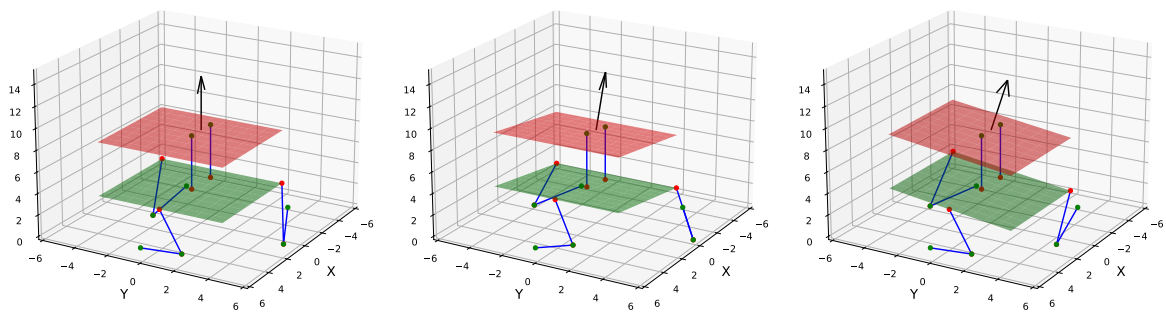


Figure 9: *(left) Collective heave variation, (centre) sin cyclic variation, (right) cos cyclic variations, all graphs showing with 10 degrees input.*

With Figure 9 (left) varying collective heave θ_h, Figure 9 (centre) varying sin cyclic θ_s and Figure 9 (right) varying cos cyclic θ_c. This Python code was transferred to C++ with the matrices expanded, and it is critical that this implementation is identical to the designed one.

To verify this, an array of values was generated in the swash plate operating range as seen in Figure 10. This array was generated by required swash plate plane angles confirming to three different frequency and amplitude sin graphs, and it was checked that the C++ calculated the same array. The chosen servos can only take an integer assignment, which limits the accuracy of the commanded plane angle.

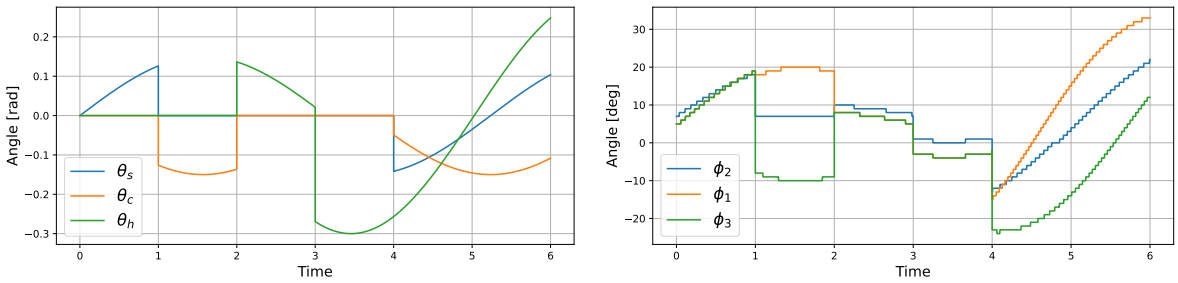


Figure 10: *(left) Controller requested swash plate angles, (right) calculated commanded servo angles.*

3 Controls Design

The next step to achieve stable flight is designing the controller. There are two main options for this controller, it could comprise of direct state feedback and pole placement. Or alternatively, decouple the states and feed these back through a nested controller structure. The advantages of each method are discussed in [20]. State-space feedback is desirable as the system's natural poles can be placed exactly in the desired location. However, this requires a very strong confidence in the model and, as such, is susceptible to inaccuracies when strong confidence in the model is not achieved. Instead, when a robust controller is desired with a weaker confidence in the model, the nested controller structure should be used. It was chosen to use a nested controller structure comprising proportional integral derivative (PID) and proportional derivative (PD) controllers similar to [2].

3.1 Plant Linearisation and Poles

The first step to designing any linear controller is linearising the plant. The standard linear state-space form is given in:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}, \quad \boldsymbol{y} = C\boldsymbol{x}, \tag{24}$$

where A is the state matrix, B is the input matrix and C is the output matrix. To find these matrices the closed equations of motion are input into a MATLAB Simulink model as shown in Figure 11 (left). The model can then be linearised in MATLAB using the using the 'linmod'

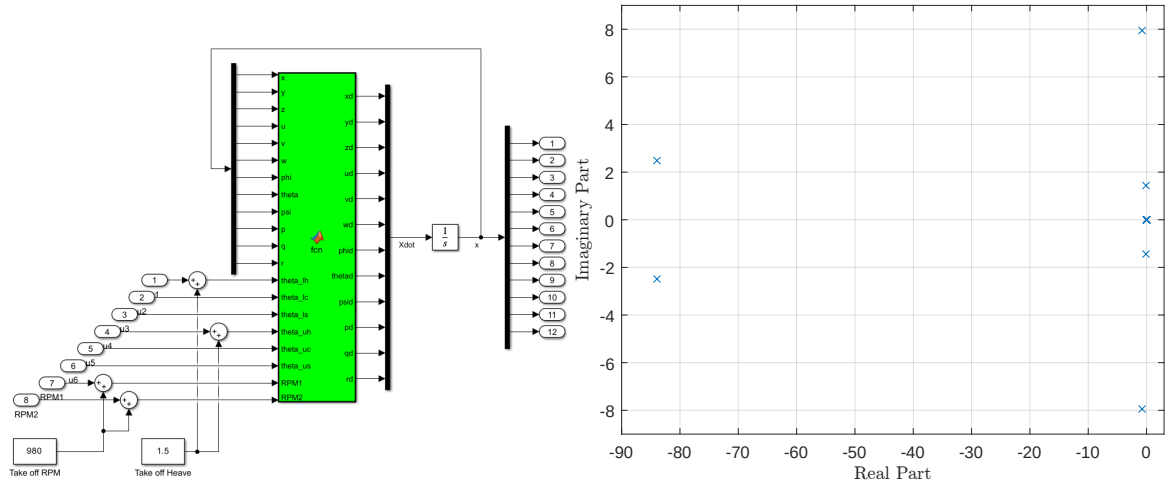


Figure 11: *(left) Open loop MATLAB Simulink system model, (right) poles of the open loop system.*

command. This directly outputs the A, B, C matrices as given by Equation (24), these are shown in Appendix F. The open loop stability of the system is assessed by finding the open loop poles as shown in Figure 11 (right). It is seen that the open loop system is marginally stable with the majority of the poles in the negative real axis, and a few open loop zeros along with some oscillatory poles very close to zero. This follows the expectation of helicopters outlined in [8] and follows intuition of the system being similar to a pendulum. It is hence necessary to include closed loop control in the system to shift the poles into the negative real axis for stable flight.

3.2 Full Closed Loop Controller

The closed-loop controller model works by feeding back the 6 states $x, y, z, \phi, \theta, \psi$ and controlling these 6 states. This works in a nested PID structure. For the height (z) and yaw (ψ) controllers the states are directly fed back into the controller. However the positions states (x, y) are fed through a nested structure into the outer block first. The pitch and roll states (ϕ, θ) are then fed through into the inner loop of this overall position loop. The input to this inner loop are calculated from the yaw mixing block. This inner loop is only PD, as there is already integration

in the forward path so there is no necessity to eliminate steady state errors in the commanded pitch and roll angles, only the position. The closed-loop controller simulation diagram is shown in Figure 12.

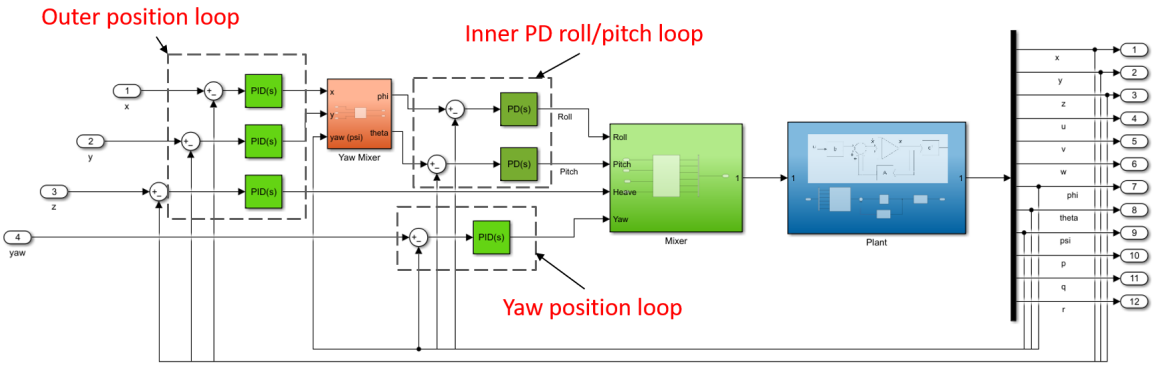


Figure 12: *Full closed loop controller MATLAB Simulink diagram.*

The input state matrix to this closed loop system can now be simplified, and rather than having it take the form of Equation (16) which included all the 8 hardware states, the input matrix can become:

$$\boldsymbol{u} = \begin{pmatrix} x & y & z & \psi \end{pmatrix}^{\top}, \tag{25}$$

where there are now only four input states. This assumes that the helicopter desires to be vertically upright[3] when stationary and is therefore sufficient to fully describe the desired helicopter position in inertial space.

3.2.1 Controls Mixing Block

The flight controller calculates a requested pitch and yaw angle along with a requested heave and yaw input to achieve the designed mission. However, the helicopter input states take the form of the hardware states detailed in Equation (16). Hence, it is necessary to go from the controller requested states to the helicopter input states. This is achieved by using decoupling matrices; these decoupling matrices are found by ensuring that the product of the decoupling matrix ($\mathbf{M}_{col}$) and the controls derivative matrix ($\boldsymbol{B}$) is diagonal. This is illustrated for the

[3] A more complicated controller could investigate holding roll and pitch angles whilst holding position states stationary, this is uniquely possible for a coaxial helicopter configuration.

collective decoupling matrix:

$$\begin{pmatrix} Z_{1h} & Z_{2h} \\ N_{1h} & N_{2h} \end{pmatrix} \cdot \mathbf{M}_{col} = \mathbf{D} = \mathrm{diag}(d_1, d_2), \tag{26}$$

where $\mathbf{D}$ represents any diagonal matrix and Z_j and N_j are the respective values in the controls matrix ($\boldsymbol{B}$). This follows the method of moving from position to hardware states in [2]. This is done for the other mixing matrices, $\mathbf{M}_{rpm}$ being the rotational speed mixer and $\mathbf{M}_{cyl,l}$, $\mathbf{M}_{cyl,u}$ being the upper and lower cyclic mixers.

After the decoupling matrices have been found they can be used directly to find the input states of the helicopter from the requested states ($z_{heave}, \psi_{yaw}, \phi_{roll}, \theta_{pitch}$), as given in:

$$\begin{pmatrix} \theta_{h1} \\ \theta_{h2} \end{pmatrix} = \mathbf{M}_{col} \cdot \begin{pmatrix} z_{heave} \\ \psi_{yaw} \end{pmatrix}, \quad \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} = \mathbf{M}_{rpm} \cdot \begin{pmatrix} z_{heave} \\ \psi_{yaw} \end{pmatrix}. \tag{27}$$

Where both the heave swash plate input and motor rotation speed can be used to control the height and yaw of the system. The decoupling matrices are sized such that if the model is accurate both collective and rotational speed control should perform equal control actuation. The cyclic mixing matrices are used to find the cyclic inputs as given in:

$$\begin{pmatrix} \theta_{c1} \\ \theta_{s1} \end{pmatrix} = \mathbf{M}_{cyl,l} \cdot \begin{pmatrix} \phi_{roll} \\ \theta_{pitch} \end{pmatrix}, \quad \begin{pmatrix} \theta_{c2} \\ \theta_{s2} \end{pmatrix} = \mathbf{M}_{cyl,u} \cdot \begin{pmatrix} \phi_{roll} \\ \theta_{pitch} \end{pmatrix}. \tag{28}$$

These cyclic matrices are again sized such that each swash plate performs equal roll and pitch control. As the helicopter turns about its centre of gravity, the lower swash plate has to command larger swash plate angles to create the same roll rates. If these matrices are not sized correctly and each swash plate does request equal roll and pitch, additional stress will be experienced in the helicopters structure.

3.2.2 Yaw Mixer

The final block in Figure 12 is the yaw mixer. This takes the input of the position states, x, y and the yaw state, ψ. The block outputs a requested roll angle ϕ and pitch angle θ, feeding the

inner loop controllers. The block consist of the transformation as given:

$$\begin{pmatrix} \theta \\ \phi \\ 0 \end{pmatrix} = \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} \tag{29}$$

Effectively this block is taking the requested position and points the helicopter's thrust vector in this direction. To prevent excessive roll and pitch angles these are limited at 45°.

3.3 Model Gain Tuning and Discussion

There are two types of compensator being used in this controller being PID and PD, with the transfer functions taking the form:

$$C_{PID}(s) = K_p + \frac{K_i}{s} + K_d \cdot s, \qquad C_{PD}(s) = K_p + K_d \cdot s, \tag{30}$$

where s is the Laplace variable and K_p, K_d, K_i are the proportional, derivative and integral controller gains. Hence there are three variables to be tuned for the PID controller and two for the PD controller. This was achieved in an iterative process following the Ziegler Nichols tuning principle [20].

3.3.1 Pitch and Roll Controller

Firstly, the inner loop pitch and roll controllers were tuned. The final tuned response is shown in Figure 13 (left). As both the pitch and roll axes are coupled, they were iteratively tuned together with a requested roll of $\theta = 0.4$ at $t = 0$ and pitch $\phi = -0.4$ at $t = 2.5$.

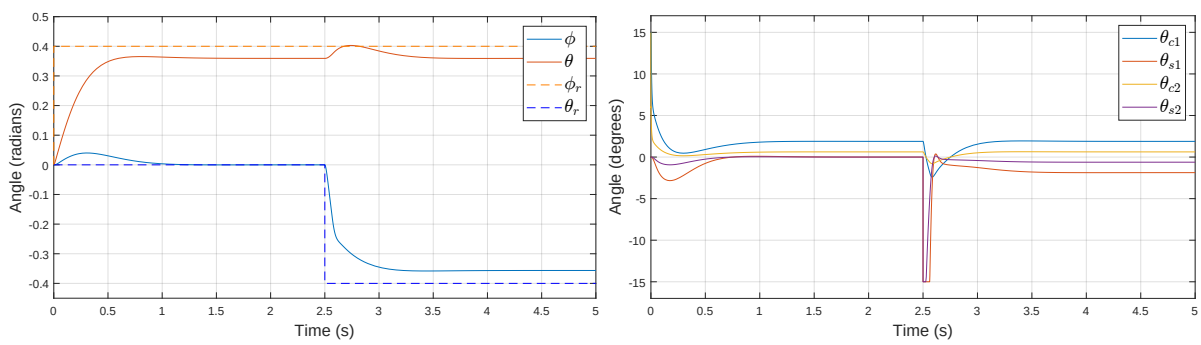


Figure 13: *(left) Pitch and roll step response, (right) commanded cyclic angles.*

As expected with PD controllers there is a steady state error; however, this is not of a concern as these inner loop controllers are fed by values from the outer position loop which already has integration, eliminating steady state error for the overall requested position. Figure 13 (right) shows the cyclic swash plate inputs required to actuate these pitch and roll angles. Here it is seen that for the first 2.5 seconds the sin cyclic control is fighting to hold zero whilst the cos cyclic holds the roll angle. Interestingly the upper swash plate has lower requested cyclic angles, this is expected with this rotor being further from the centre of gravity, hence requiring lower input to actuate the same body angles.

3.3.2 Heave Controller

Next the heave controller is tuned, which is shown in Figure 14 (left) and is in a separate loop to the pitch and roll controllers. It is desirable to have the height controller critically damped, $\zeta < 1$ such that there is no overshoot. This prevents hard landings but results in slower responses; alternatively, a faster response could be achieved with overshoot and have separate landing gains preventing hard landings. Figure 14 (centre) shows the collective plane

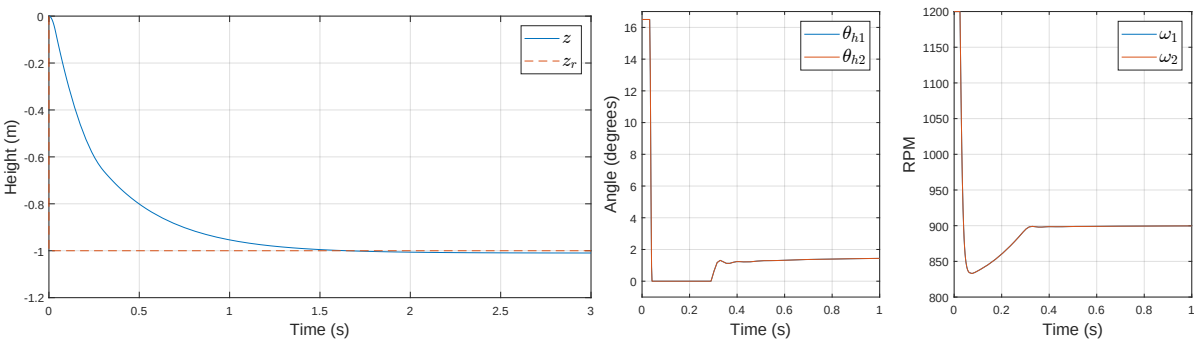


Figure 14: *(left) Height step response, (centre) commanded collective angles, (right) commanded motor rotational speed.*

angles requested. Here actuator saturation is visible, where the controller cannot command negative cyclic angles as negative lift causes singularities in the model. Figure 14 (right) shows the requested motor rotational speeds which are both equal as the model is commanding no yaw.

3.3.3 Yaw Controller

Next the yaw controller is tuned as shown in Figure 15 (left). The helicopter has very low inertia in this axis, and hence the response is very fast and exhibits overshoot. The real implemented

controller has to work hard to stabilise this axis due to the very low inertia and fast response.

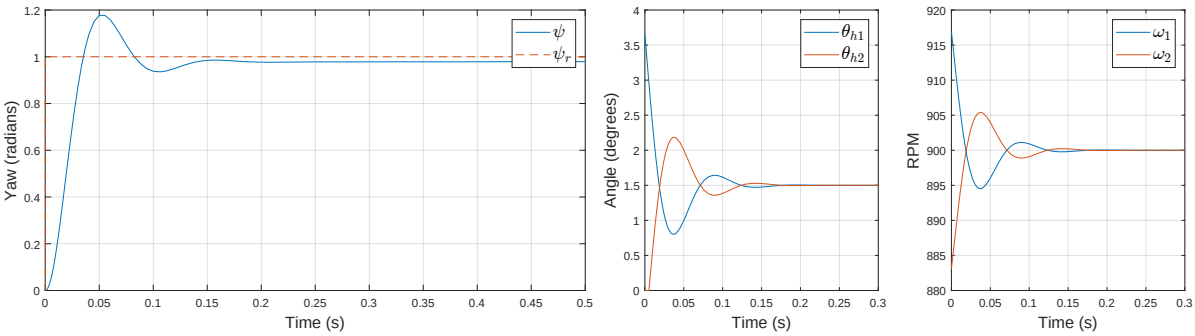


Figure 15: *(left) Yaw step response, (centre) commanded collective angles, (right) commanded motor rotational speed.*

Figure 15 (centre) shows the requested collective angles. In order to command this yaw there is a larger collective input on the upper swash plate and lower input on the lower swash plate. As this is purely collective control it directly corresponds to an angle of attack variation on the rotors. Increasing the angle of attack increases the drag and creates a torque on the helicopter. This yaw controller is equally assisted by the rotational speed variation shown in Figure 15 (right).

3.3.4 Position Controller

Finally the outer position loop is tuned and the response is shown in Figure 16 (left). This outer position loop feeds the requested pitch and roll angles. Effectively pointing the helicopter in the desired direction. It was chosen to have the position critically damped to prevent overshoots and oscillations whilst achieving a fast response.

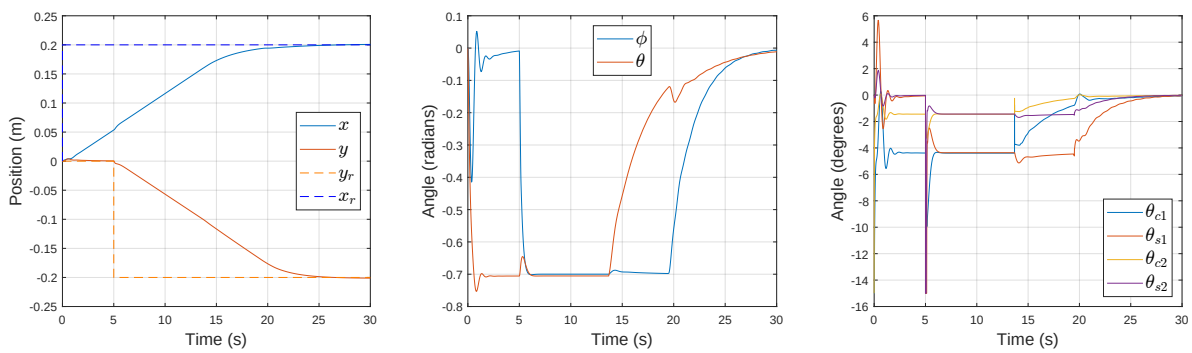


Figure 16: *(left) Position step response, (centre) commanded pitch and roll angles, (right) commanded cyclic angles.*

Figure 16 (centre) shows the requested roll and pitch rates for the commanded position. These roll and pitch angles point the helicopter's thrust vector directly towards the requested position.

Interestingly the first response of the pitch and roll to this step input exhibits overshoot. This initial overshoot corresponds to moving the helicopter from stationary to moving, and therefore is not of concern. However when reaching the requested position the pitch and roll angles slowly decrease corresponding to the critically damped position request.

3.4 Closed Loop Gains and Discrete Poles

The gains that produced this response are shown in Table 3 and are what was used for a first attempt on the real helicopter's flight tests.

Table 3: *Showing the final tuned gains.*

Controller	Proportional (K_p)	Derivative (K_d)	Integral (K_i)
Roll	3.49	0.713	-
Pitch	3.5	0.7	-
Yaw	0.0002	0.0225	0.005
Heave (z)	3	1.16	0.1
Position (x)	25	22	0.02
Position (y)	35	23.5	0.03

Although each individual system is tuned for stability, it is necessary to confirm stability of the overall system. The overall closed loop system is linearised and the poles are shown in Figure 17 (left). All poles are now on the negative real axis indicating stability of the system although some poles are very close to zero, which is undesirable. An improved controller would move these poles further into the stability region. As the actual controller will be implemented in discrete form, the discrete poles are also found in the z plane from $z = e^{sT}$. As all of the poles in Figure 17 (right) are within the unit circle, the discrete system is also expected to be stable.

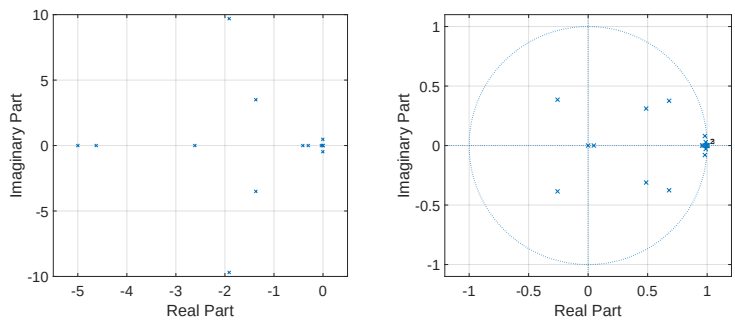


Figure 17: *(left) Closed loop poles, (right) discrete closed loop poles.*

4 Flight Software

After the underlying controller was designed and tuned, it was necessary to programme the flight software to perform this control. It was chosen to write this flight software in C++ due to the speed of execution and ease of writing and debugging. The software works by sending mode commands from a Bluetooth ground station, and automatic mode transitions from the helicopter's flight computer. All flight modes comprise of a loop executed at 119 Hz. The actual written loop had six core functions, as summarised by:

1. `Read_states();` - Read all the sensor states.

2. `Controller();` - Discrete control from requested states and current states.

3. `Kinematics();` - From the swash plate plane angles, calculate the servo angles.

4. `Hardware_assign();` - Assigns servo angles and throttle values.

5. `Right_data();` - Saves all the data.

6. `State_safety();` - Checks all states are within safety limits.

4.1 Software State Machine

The software mode commander is shown in Figure 18. This shows the standard transitions in black, the secondary path possible in blue upon fault detection, and the emergency transitions possible in red. These modes are transitioned either (*transitioned on command*) from the ground station or (*auto transition*) performed by the software autonomously. The states can be summarised as, **Idle**, the default power on software state, no hardware is armed. **Estimate States** (*transitioned on command*), the hardware is armed and the system checks that the sensor data is stable and healthy. **Ready to Takeoff** (*auto transition*), All systems are healthy and ready to take off. **Takeoff** (*transitioned on command*), the motors spin up and a neutral heave is set. **Climb, Hover, Move, Descend, Land** (*auto transition*), desired position is set and closed-loop control activates, transitions now occur on time or requested position being met. **Emergency Stop** *(auto transition/transitioned on command*) The hardware is stopped and

protected. **Spindown**: (*auto transition*), hardware is allowed to spin down and is then safed. At all points throughout these mode transitions, the helicopter is autonomously checking the system's health; if anything is found abnormal, the helicopter will either transition immediately to the landing modes or into emergency stop depending on the fault.

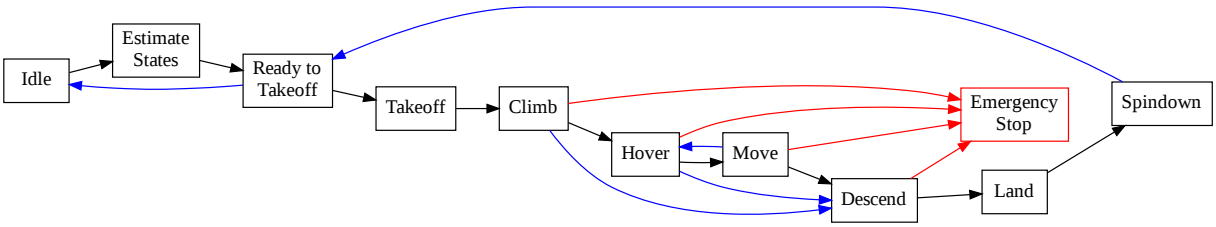


Figure 18: *Software state machine. Black is standard path, blue is secondary option and red is for emergency transitions.*

4.2 Magnetometer Calibration

The yaw, roll, and pitch states are estimated with the Madgwick library [18], which is a filter that effectively combines magnetometer, gyroscope, and accelerometer data. All of these sensors should be calibrated for best readings, but it is critical to calibrate the magnetometer. This was done by transitioning the software to a separate calibration state where the raw magnetometer readings in microtesla μT are sent over serial communication. This raw calibration data is shown in Figure 19 (left).

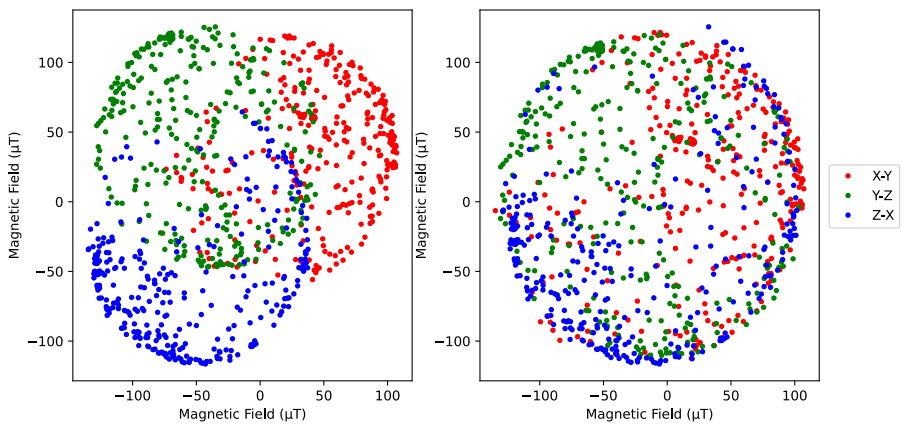


Figure 19: *(left) Raw magnetometer data, (right) calibrated data.*

This calibration data was achieved by rotating the helicopter around in all axes until circles formed on each axis. It is critical that the centres of each circle are located about $x, y = 0$. Hence, each circle is translated to achieve this; this translation is called "Hard Iron Offset", and the results are shown in Figure 19 (right). More sophisticated calibration techniques could ensure all axis result in perfect circles which are co-located but is not appropriate for real-time control.

4.3 Discrete Controller Software Implementation

The controller transfer function shown in Equation (30) must be implemented in a discrete logic form on the flight computer. The backwards difference approximation was chosen to achieve this. The backwards difference substitution takes the form:

$$s = \frac{z-1}{zT}.$$

This kept the discrete controller expressions as simple as possible whilst maintaining stability. The discrete expressions for the PID controller then become:

$$u_k = e_k \left(k_p + TK_i \frac{K_d}{T} \right) + e_{k-1} \left(-K_p - 2\frac{K_d}{T} \right) + e_{k-2} \left(\frac{K_d}{T} \right) + u_{k-1}, \tag{31}$$

where u_k is the controller plant signal and e_k is the error signal which is the difference between the requested and the output signal. T is the controller sample time and k is the discrete time index. Then the discrete expression for the PD controller takes the form:

$$u_k = \left(K_p + \frac{K_d}{T} \right) e_k - \frac{K_d}{T} \cdot e_{k-1}.$$

5 Flight Testing Results and Discussion

The final part of this project was to test the built helicopter with the designed controller. The helicopter's software was tested in an incremental process, by testing system-by-system and slowly building up to the full flight test. Firstly, the mode commander was tested without the blades installed. This allowed for verification that all systems were functioning correctly and that data acquisition was possible. Upon completion of this, the controllers were independently tested verifying stability in each individual axes.

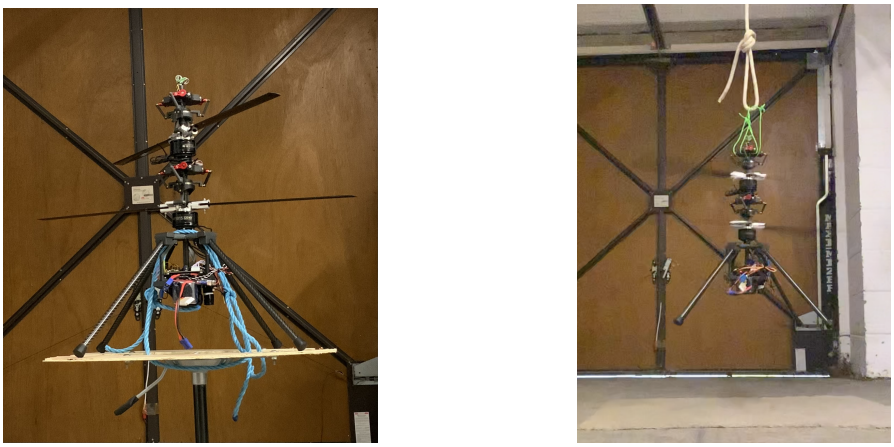


Figure 20: *(left) Yaw test stand with helicopter restrained, (right) suspended and full flight test setup with safety rope.*

5.1 Yaw Testing Stand Tests

Firstly a yaw test stand was constructed and the helicopter was strapped to this as shown in Figure 20 (left). This stand allowed the controller to be tested purely in yaw, which allowed for verification that the controller was functioning correctly and that no errors had been made in the software loops. The yaw response on this test stand is shown in Figure 21 (left). The response on this test stand was very desirable with limited steady state error and a fast settling time. Limited hardware saturation was present allowing the controller to actuate hardware to required position for the majority of the test, only in the initial transient period was saturation present. However the test stand itself introduced additional system mass and inertia in the z axis. As the helicopter inertia in this axis is relatively low, this additional inertia acted to damp the system's response. Hence although this test verified that the system was controllable, it did

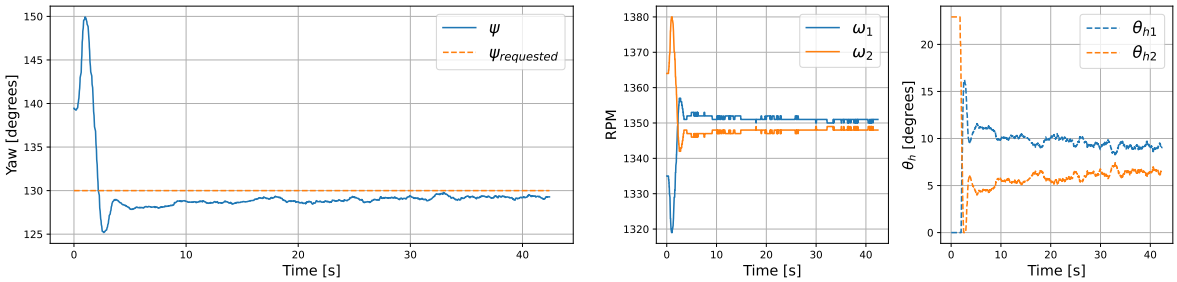


Figure 21: *(left) Yaw response on the test stand, (centre) motor rotational speed, (right) cyclic angles.*

not verify the controller gains and additional testing was required.

5.2 Suspended Yaw Controller Testing

After this desirable response was achieved on the yaw test stand, it was necessary to test the yaw controller under flight conditions. This was achieved by suspending the helicopter with the use of a rope as seen in Figure 21 (right). This test setup introduced negligible inertia in standard operating conditions; however, if the helicopter encountered excessive yaw it would get tangled in the rope causing issues. This intense yaw was common on motor start-up because there is a small transient period in which one motor spins while the other is starting up. This would result in large yaw movements and likely test issues. To counteract this, the helicopter is started up on the ground, and then the rope is used to suspend the helicopter when the software transitions to closed-loop flying modes. This setup then allowed for testing of the yaw controller independently in a method that is very similar to a full flight test. Initially, significant instabilities were observed when the simulated model gains were used for this test. This is likely because the predicted inertia from the CAD software were inaccurate. As the relative magnitude is low, this inaccuracy between the simulated model and the built model is enough to cause instabilities.

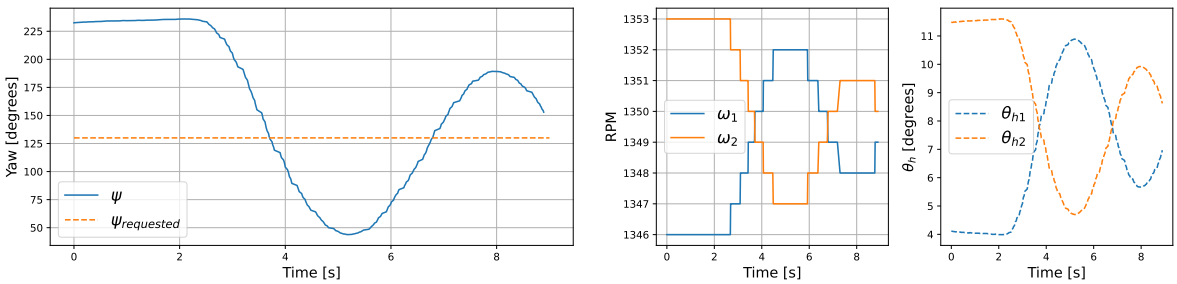


Figure 22: *(left) Proportional controller yaw response with helicopter suspended, (centre) motor rotational speed, (right) cyclic angles.*

Hence, additional controller tuning was required on the built helicopter. The same procedure adopted to tune the model's gains was adopted to tune the real helicopter's gains. Firstly, the critical yaw gain was found as shown in Figure 22. It is clearly seen in Figure 22 (centre/right) the hardware oscillations trying to counteract this yaw movement. Using this critical gain the desired PID constants are found. These tuned controller gains resulted in the response shown in Figure 23.

The final response is acceptable but undesirable. The yaw axis is stabilised but exhibits a significant steady state error. This steady state error is likely due to the actuation rate saturation. It is seen in Figure 23 (centre/right) the rapidly changing requested states. Further analysis is needed to model and incorporate this time delay into the control system. However, it is unlikely that the actuators are able to respond this fast. It is likely that with more sophisticated models that includes this actuator time delay a larger integral and derivative gain would be necessary with smaller proportional gain. Alternatively, faster responding actuators could be selected with greater positional resolution to maintain the current gains.

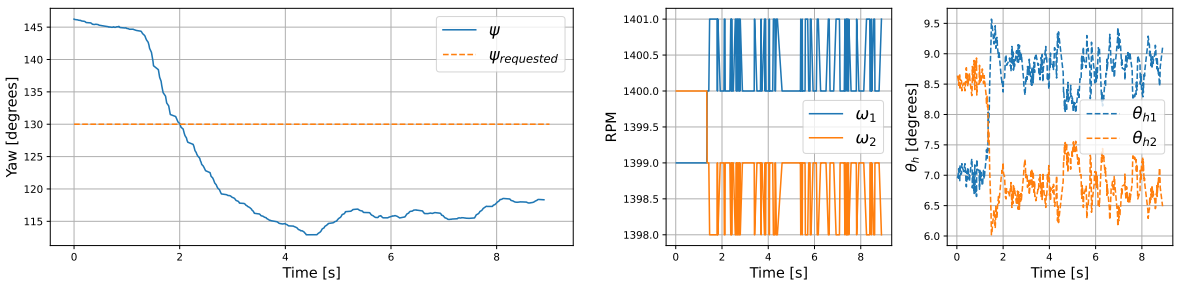


Figure 23: *(left) PID controller yaw response with helicopter suspended, (centre) motor rotational speed, (right) cyclic angles.*

5.3 Heave Controller Testing

The heave controller is then enabled and configured. In the same manor as the yaw controller, the modelled gains were not directly appropriate, and the gains had to be tuned directly on the helicopter. When using purely proportional control it is very difficult to select a gain that results in sustained oscillations whilst proving enough gain for takeoff. This is a situation where gain substitution could be used, having take-off gains and flying gains. The best response found with this purely proportional height controller is shown in Figure 24. Height oscillations begin 3 seconds into the test and 4.5 seconds in the height exceed the acceptable limit, and the test

was aborted. Nevertheless, this proportional gain was used as the critical gain to find the PID controller gains.

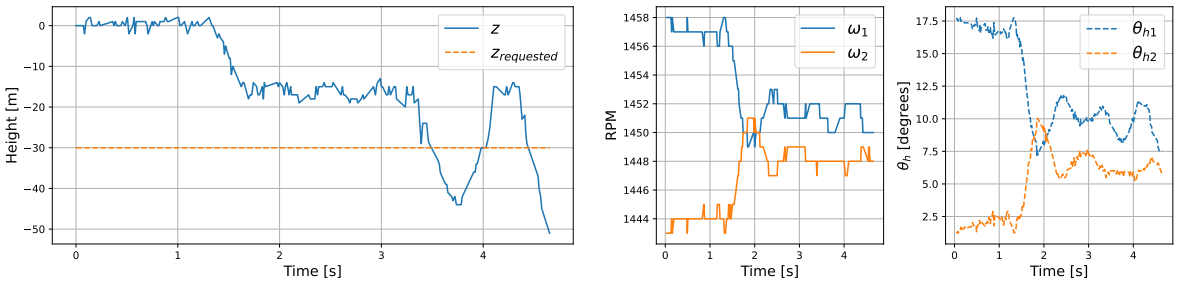


Figure 24: *(left) Proportional controller yaw response with helicopter suspended, (centre) motor rotational speed, (right) cyclic angles.*

Due to the issue of requiring large gains to liftoff, whilst simultaneously larger gains make the heave controller unstable, it is very desirable to implement PID control. Having the integral term specifically allows the plant input to grow with time allowing for smaller proportional gain. This lower proportional gain (error driving) can prevent instability whilst then still enabling take off.

5.4 Pitch, Roll and Position Controllers

The pitch and roll controllers are only necessary for changes in position by commanding the pitch and roll angles. However, the built helicopter cannot accurately determine these position states from the double integration of the accelerometer data. This double integration introduces significant noise into these readings, making them unusable by the controller. Additionally, as the helicopter was found to be open-loop stable in roll and pitch, the controllers only improve this response and do not stabilise these axes. It was not possible without a complicated test stand setup to test these controllers independently. Due to these challenges and time pressures to move towards full flight testing, it was chosen to disable these controllers.

5.5 Full Flight Testing

Finally the full flight test was attempted. This test was programmed to last for ten seconds before transitioning into the landing sequence. A height of $z = -0.3$ m was requested along with a yaw of $\psi = 340°$. Although the pitch and roll controllers were not enabled, pitch and roll states are expected to decay to a steady value upon liftoff. The data recorded from this full

flight test sequence is shown in Figure 25 (left). With the hardware states shown in Figure 25 (centre/right).

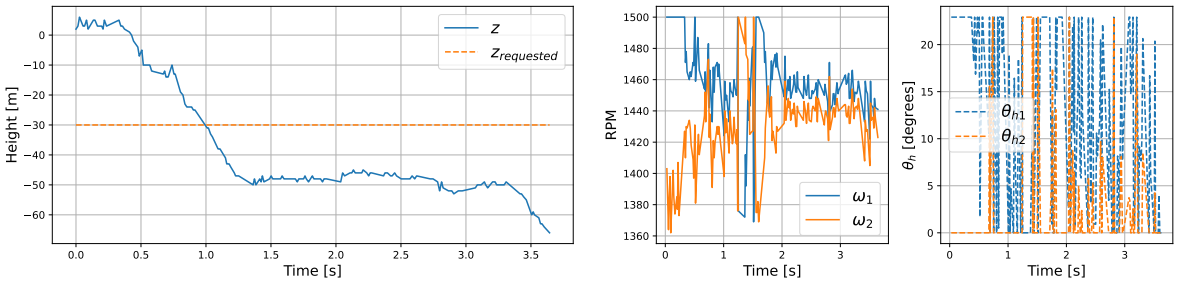


Figure 25: *(left) Proportional controller yaw response with helicopter suspended, (centre) rotational speed, (right) cyclic response.*

Although the test was programmed to last 10 seconds, 4 seconds into the test the helicopter got tangled in the safety rope. This was caused by excessive yaw movement at take-off (around 100°) and then a steady yaw drift. After becoming tangled, a sudden pitch rate outside of the allowable bounds was observed. This resulted in the helicopter autonomously transitioning into the emergency stop sequence instead of executing a full landing sequence. Regardless, the test time was adequate to observe steady state flight and understand the overall controllers' performance.

In this test a significant steady-state height error was seen, this could be eliminated by changing the PID gains. Firstly, the integral gain should be increased; however, this may result in oscillations. If this is the case, proportional control should be decreased and derivative control could be increased. Gain tuning in the flight article like this is highly iterative and dangerous, hence a better approach would be to update the model and tune it there. In this update to the model, it is important to include more accurate properties of mass and inertia.

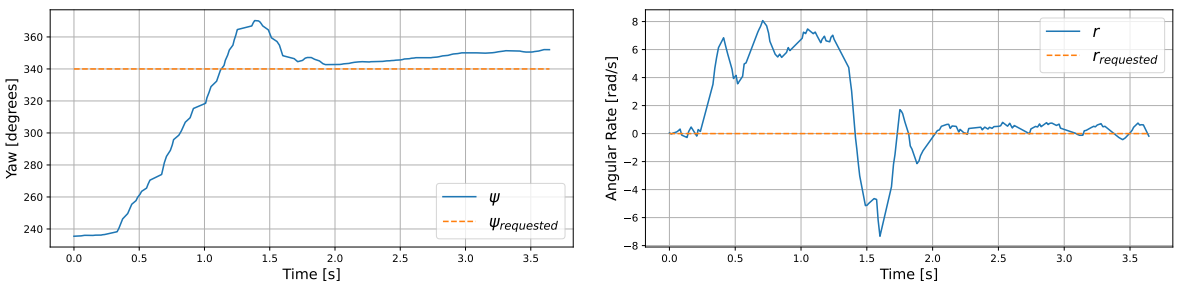


Figure 26: *(left) PID controller yaw response with helicopter suspended, (centre) rotational speed, (right) cyclic response.*

The yaw response is shown in Figure 26 (left). This yaw response was undesirable and slowly grew away from the requested angle. In addition, the rapid yaw after takeoff to the requested angle is undesirable; future software should read the current yaw at takeoff and hold this value. Fundamentally, this yaw response is what resulted in the helicopter becoming tangled and the test aborting. A key recommendation for future controllers is that rather than using Equation (31) to numerically calculate the derivative, these derivative states should be fed back directly. The yaw rate r is measured directly and is shown in Figure 26 (right). Feeding back on this should result in a quicker and more accurate response compared to finding these states numerically in the discrete controller approximation given in Equation (31). The angle states that are inputted into this approximation have already passed through a low-pass estimation filter potentially attenuating the fast-changing velocity shown in Figure 26 (right).

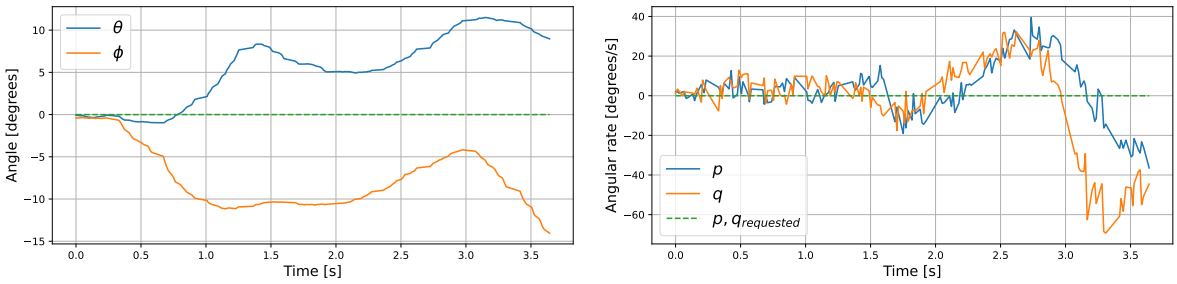


Figure 27: *(left) PID controller pitch and roll response, (right) pitch and roll rates.*

Finally, the roll and pitch states are observed in Figure 27 (left). These are stable but exhibit some bounded oscillations; this would be improved by enabling the roll and pitch controllers. However, as these controllers performance could not be verified independently it might also result in unstable oscillations, and hence it was safest to disable them. In the same recommendations as for the yaw controller, it is also recommended to feed back on the roll and pitch rates directly, as shown in Figure 27 (right). Furthermore, additional testing should independently verify pitch and roll controllers prior to future flight attempts.

6 Conclusion

The objectives of this project were to design, manufacture and test a coaxial helicopter, achieving stable hovering on earth. Significant time was spent redesigning the helicopter for earth operation; however, the focus of this report was on the modelling and control of the helicopter. The redesign included updating the swashplates geometry, blade grips, blades, main body, battery mount, sensor mounting, motor mounting, shaft, shaft hubs and manufacturing a flight controller PCB.

The helicopter controller was designed and simulated in Matlab Simulink. The open-loop system was linearised, and the system poles were found to be marginally stable, hence requiring closed-loop control for stable flight. The six controllers were tuned with step responses. The inner pitch and roll loop PD controllers exhibited a steady-state error as expected without integral control. The overall position loop that fed the inner loop were PID controllers and eliminated any positional error. The height controller was tuned to be critically damped, this ensured that soft landings were possible. Alternatively, gain substitution could be used and have separate flying and landing gains. Finally, the yaw controller was tuned, the response was very fast but had substantial steady-state error. This axis was very susceptible to the helicopter inertia, and hence small errors between this and the built model would cause issues. The entire controller was linearised, and the closed-loop poles were found, these were all in the stable region. These poles were then put into the discrete domain and were also all stable.

Flight software was written that implements the designed controller. This flight software included robust safety systems capable of aborting the helicopter out of test sequences if it detected errors. This flight software and the aborting system worked successfully in many test flights.

Finally, flight tests were conducted. Each axis was independently tested before a full flight test. The final flight test demonstrated a stable flight for 3.5 seconds before the helicopter hit the safety rope resulting in an excessive roll movement and the software aborting out of the test. This stable flight was considered a success; however, the functionality of the controller could be improved to reduce steady-state errors. Increasing the integral controller gains should reduce

this error, although integral gain is likely to make the system increasingly unstable, hence it was decided to pause flight testing. It is recommended prior to further testing that the simulated model be improved and the roll and pitch controllers should be verified such that these can be enabled.

In conclusion, the helicopter is very close to achieving sustained hover. It is recommended that future work focus on the yaw and roll controllers, potentially selecting new servos with faster actuation time. Additionally, work should be put into the swash plate design to provide a better commanded accuracy of the plane angles with a greater servo travel required. Furthermore, the simulation model can be improved by updating helicopter properties more accurately or potentially by linearising directly from flying tests rather than the equations of motion.

References

[1] Jansen Papworth et al. *Mars Helicopter - Propulsion Systems*. Imperial College London, ME3 Design, Make and Test, DMT Group 04-B. 2023.

[2] Bob Balaram et al. *Mars Helicopter Technology Demonstrator*. Atmospheric Flight Mechanics Conference. 2018. DOI: 10.2514/6.2018-0023.

[3] Puneet Singh. *Aeromechanics of Coaxial Rotor Helicopters using the Viscous Vortex Particle Method*. The University of Michigan. 2020. DOI: 10.7302/19z4-3f29.

[4] Constantin Rotaru et al. *Helicopter Flight Physics*. IntechOpen, 2018. DOI: 10.5772/intechopen.71516.

[5] NASA. *Ingenuity Mars Helicopter*. https://science.nasa.gov/mission/mars-2020-perseverance/ingenuity-mars-helicopter. Accessed: 21 May 2024.

[6] Thomas Kraeutner. *The Airfoil Method for Rotary Wing Analysis Using Modified Lift and Drag Equations*. Tinker AFB. 2021.

[7] Guowei Cai et al. *Construction, Modeling and Control of a Mini Autonomous UAV Helicopter*. IEEE International Conference on Automation and Logistics. 2008. DOI: 10.1109/ICAL.2008.4636193.

[8] Wayne Johnson. *Helicopter Theory*. New York, NY, USA. Dover Publications, 1980.

[9] J. Gordon Leishman. *Principles of Helicopter Aerodynamics*. Cambridge, UK. Cambridge University Press, 2006.

[10] Jianbo Liu et al. *A Novel Comprehensive Kinematic and Inverse Dynamic Model for the Flybar-Less Swashplate Mechanism*. Symmetry. 2020. DOI: 10.3390/sym12111834.

[11] Tobias Richter et al. *System Identification of a Coaxial Ultralight Helicopter*. University of Stuttgart, Germany. 2023. DOI: 10.4050/F-0076-2020-16288.

[12] C. Venkatesan. *Fundamentals of Helicopter Dynamics*. Boca Raton, FL, USA. CRC Press, 2014.

[13] Gareth D. Padfield. *Helicopter Flight Dynamics, 3rd Edition: Including a Treatment of Tiltrotor Aircraft*. New York: Wiley, 2018.

[14] Gareth D. Padfield. *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*. Hoboken, NJ, USA. Wiley-Blackwell, 2018.

[15] A.R.S. Ramwell. *Helicopter Dynamics*. London, UK. Edward Arnold Publications, 1976.

[16] MathWorks. *Rotor Dynamics (Aerospace Blockset)*. https://uk.mathworks.com/help/aeroblks/rotor.html. Accessed: April 30, 2024.

[17] UIUC Applied Aerodynamics Group. *Eppler E169 Low Reynolds Number Airfoil*. http://airfoiltools.com/airfoil/details?airfoil=e169-il. Accessed: 9th February 2024.

[18] Sebastian O.H. Madgwick. *An efficient orientation filter for inertial and inertial/magnetic sensor arrays*. University of Bristol. 2010.

[19] Mohammad Reza Sabapour and Hassan Zohoor. *Analysis of a Swashplate Mechanism of the Hingeless Rotor Hub with the Flybar in a Model Helicopter, Part I: Kinematics*. 2010. DOI: 10.1299/jsdd.4.616.

[20] J. P. Hespanha. *Linear Systems Theory: Second Edition*. NED-New edition, 2. Princeton University Press, 2018. DOI: 10.2307/j.ctvc772kp.

Appendices

A Swash Plate and Blade Grip Development

The newly designed blade grips and swash plate are shown in Figure 28 (left). The blade grip design includes two ball bearings and one thrust bearing. This appropriately restrains the blades. The new swash plate design correctly constrains the stationary and rotational parts of the swash plate. The manufactured helicopter swash plate and blade grips is shown in Figure 28 (right) without the blades installed. This was the configuration used for initial software and hardware testing.

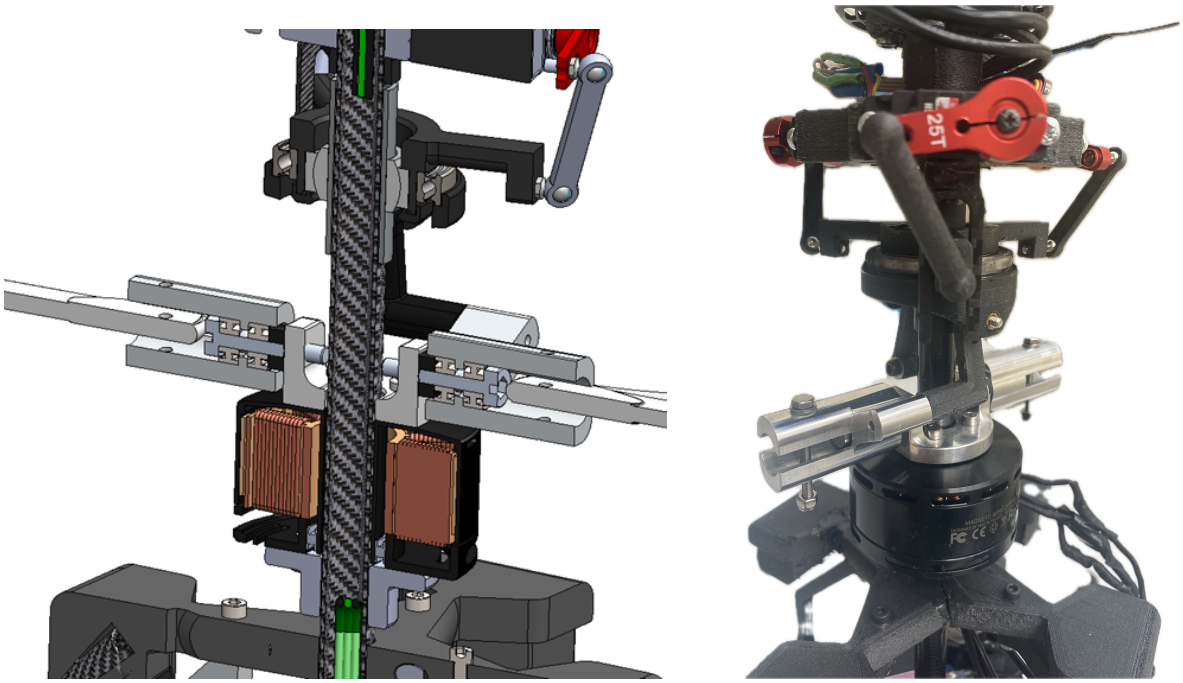


Figure 28: *(left) Designed CAD swash plate and blade grips cross section, (right) manufactured blade grips and swash plate.*

B Electronic Hardware Development

The final helicopters electronic hardware can be summarized below:

- *Garmin LIDAR-Lite* v3 - Measuring vertical inertial height .
- 40A 6 *Cell Electronic Speed Controllers* - Controlling the motors rotational speed.)
- *Arduino Nano* 33 *BLE (Containing* LSM9DS1 9 *axis IMU)* - Measuring eulur angles, accelerations and the translation accelerations whilst communicating with the ground station over Bluetooth.

It was necessary to manufacture a PCB due to complicated power arrangements and significant amount of sensors and cables. The engineering drawings for this PCB is shown in 29.

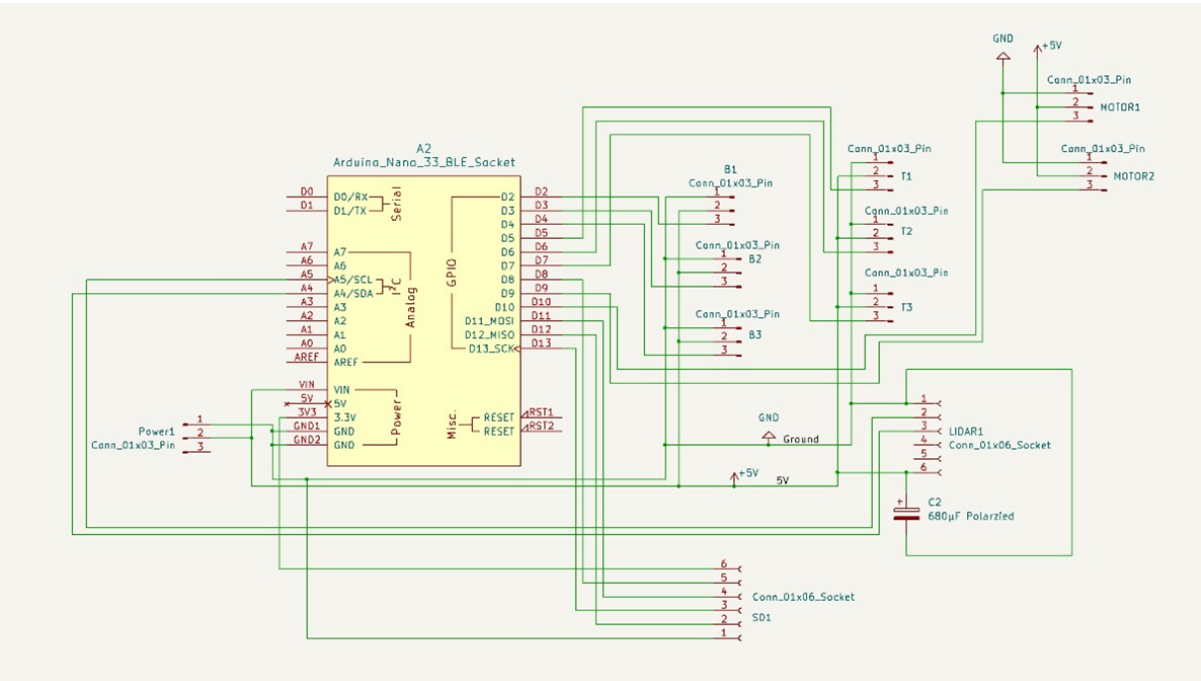


Figure 29: *Engineering drawing of the flight controller PCB design.*

The manufactured board is shown in Figure 30 (left) and the board integrated into the helicopter is shown in Figure 30 (right)

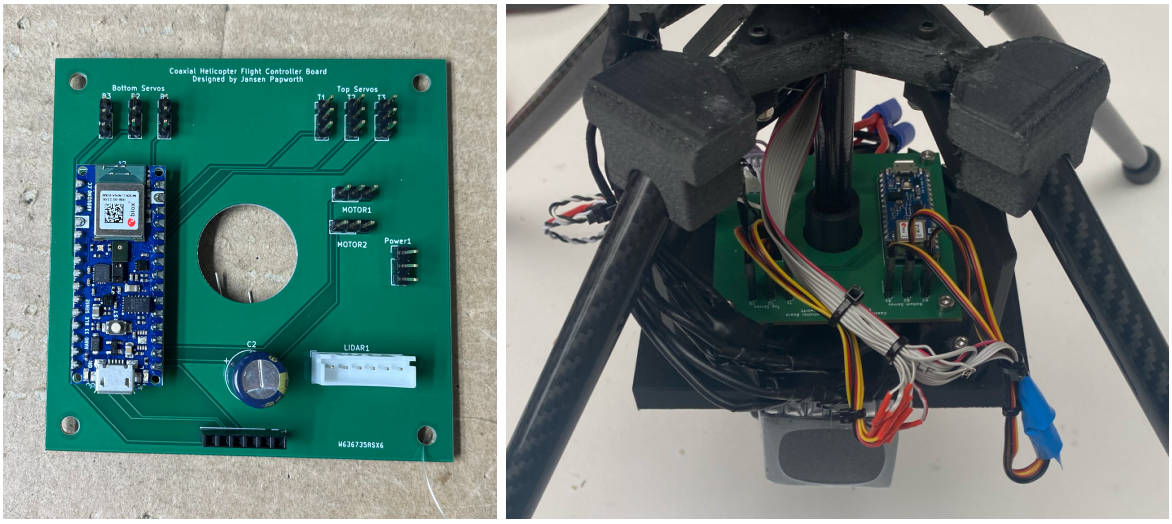


Figure 30: *(left) Manufactured PCB. (right) Manufactured PCB integrated into the assembled helicopter*

C Flapping Coefficients

The linear flapping coefficients are listed in equation (32) to (43).

$$\beta_{c1,u} = -8\left(\frac{2C_T}{\sigma a} + \frac{1}{4}\sqrt{\frac{C_T}{2}}\right) \tag{32}$$

$$\beta_{c1,q} = \frac{16}{\gamma}\left(\frac{2C_T}{\sigma a} + \frac{1}{4}\sqrt{\frac{C_T}{2}}\right) h_1 \tag{33}$$

$$\beta_{c1,p} = 1 \tag{34}$$

$$\beta_{s1,v} = -8\left(\frac{2C_T}{\sigma a} + \frac{1}{4}\sqrt{\frac{C_T}{2}}\right) \tag{35}$$

$$\beta_{s1,p} = \frac{16}{\gamma}\left(\frac{2C_T}{\sigma a} + \frac{1}{4}\sqrt{\frac{C_T}{2}}\right) h_1 \tag{36}$$

$$\beta_{s1,q} = 1 \tag{37}$$

$$\beta_{c2,u} = -8\left(\frac{2C_T}{\sigma a} + \frac{1}{4}\sqrt{\frac{C_T}{2}}\right) \tag{38}$$

$$\beta_{c2,q} = \frac{16}{\gamma}\left(\frac{2C_T}{\sigma a} + \frac{1}{4}\sqrt{\frac{C_T}{2}}\right) h_2 \tag{39}$$

$$\beta_{c2,p} = -1 \tag{40}$$

$$\beta_{s2,v} = 8\left(\frac{2C_T}{\sigma a} + \frac{1}{4}\sqrt{\frac{C_T}{2}}\right) \tag{41}$$

$$\beta_{s2,p} = \frac{16}{\gamma} - 8\left(\frac{2C_T}{\sigma a} + \frac{1}{4}\sqrt{\frac{C_T}{2}}\right) h_2 \tag{42}$$

$$\beta_{s2,q} = 1 \tag{43}$$

D Thrust Coefficient Derivation

The Matlab code that was used to derive the thrust coefficients is shown Listing 1, this is based off:

$$dC_T = 4F\lambda^2 r\,dr, \quad F = \frac{2}{\pi}\cos^{-1}\exp\left(-\frac{1}{2}\frac{Nb(1-r)}{r\varphi}\right).$$

$$dC_T = \frac{1}{2}\sigma c \cdot a(\frac{\theta_t}{r}r - \lambda)rdr$$

Listing 1: *MATLAB Code*

```
N = 2; % number of blades
Chord = 0.00313; % chord length
sigma = 0.054;
a = 10; % lift curve slope

% starting points of the iteration
h = 0.001;
R = 0.714/2;
r_values = 0:h:R; % values of r
l_values = []; % inflow ratio values

theta = [0:1:10];
CT_values = [];
for theta_h = theta
    for r = r_values
      fun = @(lamda) 0.5*sigma*Chord*a*(theta_h - lamda) - ...
                4*(2/pi)*acos(exp((N*(r - 1))/(2 * lamda)))*lamda^2;

      % Initial guess (adjust if needed)
      x0 = [0.05];

      % Solve for l using fsolve
      l = fsolve(fun, x0);
```

```
        % Append the calculated l to the l_values array
        l_values = [l_values, l];
    end

    % calculating CT using the trapezium rule summation
    A = l_values(1);
    B = l_values(length(l_values));
    f_x = l_values(2:length(l_values)-1);

    CT = h * (A/2 + sum(f_x) + B/2);
    CT_values = [CT_values, CT];
end

a_t = 0.0118;
% thrust curve slope
CT_values1 = theta*a_t;
plot(theta, CT_values,theta,CT_values1)%,Marker = 'True Solution');
% let gradient = 5
xlabel('\theta_h [degrees]','FontSize', 14);
ylabel('C_T','FontSize', 14);
```

E Kinematic Transformations Matrices

The kinematics transformation are functions of $C_5^0(\theta_h, \theta_c, \theta_s)$. This is the combination of three transformation matrices as given by:

$$R_X(\beta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & \sin\beta & 0 \\ 0 & -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_Z(\alpha) = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ -\sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_z(h) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

These are then multiplied together forming the overall transformation matrix. This is used to calculate the required location of the servo arm C and is given in:

$$C_5^0(\theta_h, \theta_c, \theta_s) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta_s \cdot c\theta_c + s\theta_s \cdot s\theta_c & c\theta_s \cdot s\theta_c - s\theta_s \cdot c\theta_c & \theta_h\left(c\theta_s \cdot s\theta_c - s\theta_s \cdot c\theta_c\right) \\ 0 & -s\theta_s \cdot c\theta_c + c\theta_s \cdot s\theta_c & -s\theta_s \cdot s\theta_c - c\theta_s \cdot c\theta_c & \theta_h\left(-s\theta_s \cdot s\theta_c - c\theta_s \cdot c\theta_c\right) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

F State Matrices

The linear state space matrices A, B, C are given:

$$A = \begin{bmatrix}
0 & 0 & 0 & 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 \\
0 & 0 & 0 & 0 & 1.0000 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.0000 & 0 & 0 & 0 & 0 \\
0 & 0 \\
0 & 0 & 0 & -59.9494 & 0 & 0 & 0 & -0.1712 & 0 & -0.0000 \\
14.4472 & 0 \\
0 & 0 & 0 & 0 & -59.9494 & 0 & 0.1712 & 0 & 0 & -14.4472 \\
0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 57.2958 \\
0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
57.2958 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 57.2958 \\
0 & 0 & 0 & 0 & -95.2674 & 0 & 0 & 0 & 0 & -24.0413 \\
8.8477 & 0 \\
0 & 0 & 0 & 287.9642 & 0 & 0 & 0 & 0 & 0 & -26.7438 \\
-72.6693 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0
\end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.081 & 0 & 0 & -0.081 & 0 & 0 & 0 \\ 0 & 0 & 0.081 & 0 & 0 & 0.081 & 0 & 0 \\ -3.102 & 0 & 0 & -3.102 & 0 & 0 & -0.010 & -0.010 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.208 & 0 & 0 & 0.069 & 0 & 0 \\ 0 & 0.194 & 0 & 0 & 0.064 & 0 & 0 & 0 \\ 17.972 & 0 & 0 & -17.972 & 0 & 0 & 0.040 & -0.040 \end{bmatrix},$$

$$C = I_{12}.$$

Appendix: Evidence that this font conforms with the page limit

A student who has not used Arial font, font size 11 point, with double line spacing, must copy and paste this page, with its heading, as an appendix to their report. Report markers will use it to verify that the font used has not resulted in a content length advantage for the student. This passage of text fills exactly one page in Arial font, font size 11 point, with double line spacing. In the student's chosen font/format, it must also fill one page or more, but under no circumstances less than one page. This appendix does not count towards the 40-page limit. **Repeat 1:** A student who has not used Arial font, font size 11 point, with double line spacing, must copy and paste this page, with its heading, as an appendix to their report. Report markers will use it to verify that the font used has not resulted in a content length advantage for the student. This passage of text fills exactly one page in Arial font, font size 11 point, with double line spacing. In the student's chosen font/format, it must also fill one page or more, but under no circumstances less than one page. This appendix does not count towards the 40-page limit. **Repeat 2:** A student who has not used Arial font, font size 11 point, with double line spacing, must copy and paste this page, with its heading, as an appendix to their report. Report markers will use it to verify that the font used has not resulted in a content length advantage for the student. This passage of text fills exactly one page in Arial font, font size 11 point, with double line spacing. In the student's chosen font/format, it must also fill one page or more, but under no circumstances less than one page. This appendix does not count towards the 40-page limit. **Repeat 3:** A student who has not used Arial font, font size 11 point, with double line spacing, must copy and paste this page, with its heading, as an appendix to their report. Report markers will use it to verify that the font used has not resulted in a content length advantage for the student. This passage of text fills exactly one page in Arial font, font size 11 point, with double line spacing. In the student's chosen font/format, it must also fill one page or more, but under no circumstances less than one page. This appendix does not count towards the 40-page limit. **Repeat 4:** A student who has not used Arial font, font size 11 point, with double line spacing, must copy and paste this page, with its heading, as an appendix to their report. Report markers will use it to verify that the font used has not resulted in a content length advantage. END

Appendix: Evidence that this (figure caption) font conforms

A student who has not used Arial font, font size 11 point, with double line spacing, must copy and paste this page, with its heading, as an appendix to their report. Report markers will use it to verify that the font used has not resulted in a content length advantage for the student. This passage of text fills exactly one page in Arial font, font size 11 point, with double line spacing. In the student's chosen font/format, it must also fill one page or more, but under no circumstances less than one page. This appendix does not count towards the 40-page limit. ***Repeat 1:*** *A student who has not used Arial font, font size 11 point, with double line spacing, must copy and paste this page, with its heading, as an appendix to their report. Report markers will use it to verify that the font used has not resulted in a content length advantage for the student. This passage of text fills exactly one page in Arial font, font size 11 point, with double line spacing. In the student's chosen font/format, it must also fill one page or more, but under no circumstances less than one page. This appendix does not count towards the 40-page limit.* ***Repeat 2:*** *A student who has not used Arial font, font size 11 point, with double line spacing, must copy and paste this page, with its heading, as an appendix to their report. Report markers will use it to verify that the font used has not resulted in a content length advantage for the student. This passage of text fills exactly one page in Arial font, font size 11 point, with double line spacing. In the student's chosen font/format, it must also fill one page or more, but under no circumstances less than one page. This appendix does not count towards the 40-page limit.* ***Repeat 3:*** *A student who has not used Arial font, font size 11 point, with double line spacing, must copy and paste this page, with its heading, as an appendix to their report. Report markers will use it to verify that the font used has not resulted in a content length advantage for the student. This passage of text fills exactly one page in Arial font, font size 11 point, with double line spacing. In the student's chosen font/format, it must also fill one page or more, but under no circumstances less than one page. This appendix does not count towards the 40-page limit.* ***Repeat 4:*** *A student who has not*

used Arial font, font size 11 point, with double line spacing, must copy and paste this page, with its heading, as an appendix to their report. Report markers will use it to verify that the font used has not resulted in a content length advantage. END.